Polyphonic Music Retrieval: The N-gram Approach

Shyamala Doraisamy

Department of Computing
Imperial College London
University of London

Supervisor: Dr. Stefan Rüger

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College

September 3, 2004

Abstract

This Music Information Retrieval (MIR) study investigates the use of n-grams and textual Information Retrieval (IR) approaches for the retrieval and access of polyphonic music data. IR, synonymous with text IR, implies the task of retrieving documents or texts with information content that is relevant to a user's information need.

With music retrieval, the use of n-grams has largely been confined to monophonic musical sequences. The few studies that have investigated its use with polyphonic music collections typically reduce a polyphonic file into a monophonic sequence for n-gram construction. Techniques for full-music indexing of polyphonic music data with n-grams are investigated. A method to obtain n-grams from polyphonic music data is introduced. The information content of 'musical n-grams' is extended to include rhythmic information in addition to intervallic information. For this, ratios of onset times between two adjacent pairs of pitch events are used. To encode 'musical n-grams' to obtain 'musical words' for indexing, a function that maps interval classes to text characters is formulated, and ranges of ratio bins are defined. These encoding approaches enable encoding of the pitch and rhythm information at various levels of coarseness. Various n-gramming strategies are proposed to overcome several problems that arise from the use of the n-gram method with polyphonic music. In exploiting the time-dependent element of polyphonic music data, a method to index adjacent and concurrent musical words using a 'polyphonic musical word indexer' is proposed. For the retrieval of these 'overlaying' musical words, i.e., when more than one word can assume the same within-document position, a new proximity-based operator and a ranking function is proposed.

The evaluation results of the indexing approaches proposed are presented, performed on a test collection we developed using approximately 10,000 polyphonic MIDI files. Experiments show that different n-gramming strategies and encoding precision differ widely in their

effectiveness. The retrieval performances of monophonic and polyphonic queries made to a polyphonic music collection were investigated using text retrieval performance measures. For monophonic queries, we focused in particular on query-by-humming systems, and for polyphonic queries on query-by-example. Error models of these systems were surveyed and included in the fault-tolerance study that investigated the robustness of the n-gram method. The feasibility in utilising position information of 'overlaying' musical words was investigated using various proximity-based and structured query operators available with text retrieval systems. Results show that the n-gram approach to polyphonic music retrieval is a promising and robust approach for indexing large collections of music.

Acknowledgements

This thesis would not have been possible without the guidance and supervision of Dr. Stefan Rüger, whom I have to thank first and foremost.

I would also like to thank my co-supervisor Dr. Krysia Broda and Dr. Duncan Gillies who initially co-supervised this work. Colleagues from Imperial College London that I would like to thank for the numerous discussions include Dr. Thomas von Schroeter, and members of the Multimedia Information Retrieval research group, namely Marcus Pickering, Daniel Heesch and Peter Howarth.

I thank the MIR research community, in particular Dr. Stephen Downie, Dr. Mark Plumbley and Tim Crawford for events organised. My active participation in these events provided me with invaluable experience as a PhD student in the MIR field. I thank Drs. Downie and Plumbley once again, this time for examining my thesis and for their invaluable suggestions towards its revision.

I would like to thank the Malaysian Government for funding my studies and my employer University Putra Malaysia for the leave. Thank you to all the staff of Imperial College London, and in particular those from the Department of Computing who have helped in some way or other in making this thesis possible.

Many thanks to Sasivimol Kittivoravitkul and Jane Labadin for all the help rendered, both in and out of College.

As for my family, I think I just can't thank them all enough. Thank you to my Mum, Sathia Bhama and my husband's family for all the patience and much needed help, especially with baby-sitting duties. Greatly appreciated.

Last but not least, I need to thank the two very important persons in my life — Suresh and Thurai for tolerating the chaotic household.

I dedicate this thesis to my late father, C. Doraisamy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis begins with an introduction to the emerging research field of Music Information Retrieval (MIR). An overview of the main challenges that MIR researchers and developers face in developing content-based MIR systems are discussed. This is followed by a description of the research focus of this MIR study investigating the use of n-grams towards the development of a polyphonic music retrieval system. The problem statement, research objectives and scope of the study are outlined. This is followed by a description of the chapters to follow in this thesis.

1.1 Music Information Retrieval

Music documents encoded in digital formats have rapidly been increasing in number with the advancements in computer and network technologies. The difficulty in managing large collections of these documents has placed great demands towards the research and development of computer-based MIR systems. Information Retrieval (IR) is a field concerned with the structure, analysis and organisation, storage, searching, and retrieval of information (Salton 1968). Amongst the research and development issues of automated IR systems, when it began in the 1940s, was the management of huge scientific literature (Frakes 1992; Vickery 1994; Besterman 1945). These documents were text documents. IR, often regarded as being synonymous with document retrieval, and more recently with text retrieval, implies the task of retrieving documents or texts with information content that is relevant to a user's information need (Spärck Jones and Willett 1997). With the current advancements in mul-

timedia technology, the information content for retrieval is no longer confined to collections of text-based documents but now consists of very diverse media such as images, video and audio. Various branches of media-specific IR research fields have now emerged due to challenges that are specific to the characteristics of the various media, and the specialised domain knowledge required for developing modern-age IR systems. One such field is the field of MIR where a growing international MIR research community is being formed, drawing upon multi-disciplinary expertise from computer science, audio engineering, library science, information science, cognitive science, musicology and music theory (Downie 2003b).

The information retrieval process can briefly be described based on a general IR model as shown in Figure 1.1. Information items in a collection are preprocessed and indexed. During information retrieval, a user's query is processed and formulated to the format requirements of the indexed collection. Information items that are most similar to the query would be retrieved and presented to the user based on a similarity computation. Developments on this basic model have resulted in rather diverse IR models, but with the common general emphasis of retrieving information relevant to a request, rather than direct specification of a document (Heaps 1978). Modern IR systems include sophisticated indexing, searching, retrieving technologies, similarity computation algorithms and user-interfaces. Using music documents as information items for the development of IR systems, i.e., MIR systems, would pose many challenges.

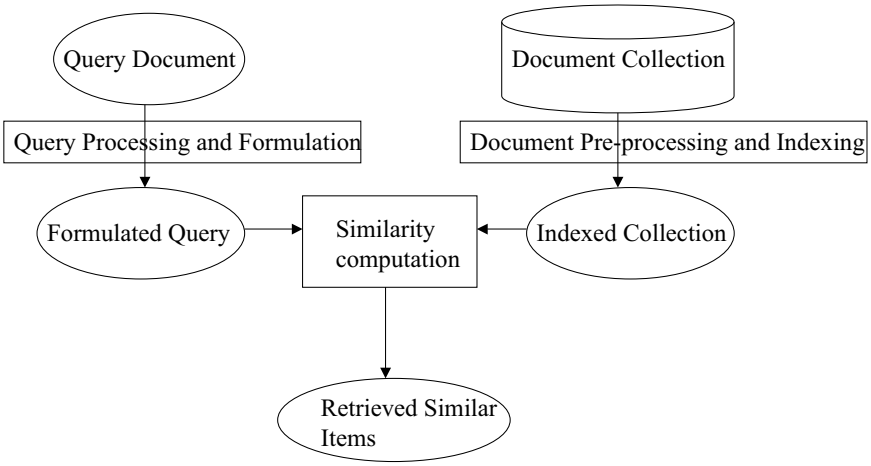


Figure 1.1: A general IR model

Music documents encompass documents that contain any music-related information such as music recordings, musical scores, manuscripts or sketches and so on (Huron 2000). Studies

have been carried out in using the music-related information contained in these documents for the development of content-based MIR systems such as Marsyas (Tzanetakis 2003), Meldex (McNab et al 1997), Themefinder (Kornstadt 1998) and Musedata (Hewlett 2001). Such systems retrieve music documents based on the information content such as the incipits, themes, genre or period in music history. However, most such content-based MIR systems developed are either experimental or research prototypes. MIR systems that are operational[1] or in widespread use are systems that have been developed using meta-data such as filename, length, title, opus number and catalogue reference. This is due to the many challenges that MIR researchers and developers face in developing such content-based systems. These challenges are presented as a brief overview in this chapter and with more details in chapters to follow.

MIR users could be highly specialised, such as music scholars, or simply be one of the millions of Web-users wishing to access pop or classical music performances on the Internet. With the former class of users a musical signature (a term for motives common to two or more works of a given composer) could be a possible example of a music query to retrieve the period of music history a work comes from or the probable composer of the work (Cope 1998). A query by melody (by humming, whistling, keyboard, etc.) to retrieve all similar pieces is a highly likely query with the latter user class (Durey and Clements 2001). Query-by-melody (QBM) is currently one of the most highly appealing form of query, not just for Web-users but most classes of MIR systems users — music librarians, disc-jockeys, music scholars, etc. (Downie 2003b; Huron 2000). However, agreement on similar pieces based on a melody which is not a simple task for humans, is simply an onerous one for computers (Selfridge-Field 1998; Byrd and Crawford 2002; Hoffman-Engl 2002). One seemingly obvious solution would be to retrieve the piece that the musical query sequence recurs the most number of times, but this may not be the case musically. In making judgements on the relevance of a music document to be retrieved for a given query, one clear challenge is addressing human musical perception. Deciding which documents are relevant given a particular query constitutes an important theoretical and practical challenge. Among the more immediate questions this raises are: would performances with the melody transposed or played in a different speed be considered

[1]Experimental IR is mainly carried out in a 'laboratory' situation whereas operational systems are commercial systems which charge for the service they provide (Rijsbergen 1979).

similar? Must performances be in the same genre? Is there a minimum to the number of times that the query melody has to be heard in the piece? Was the melody heard as part of the accompaniment? How does the computer intuitively decide which is the accompaniment? Can the melody be slightly varied? If yes, what is the degree of variation allowed? This list would just continue to include more musically in-depth questions that may even dispute the validity of the query melody. Must the query be musically complete in structure (theme, motive, phrase, fugal subject, etc.)? Is a query with an arbitrary number of notes possibly not complete musically, analogous to half a word or sentence with text? It would be difficult to define relevant documents if the query is not even considered valid! These issues are just a few of the many that would have to be addressed in order to define relevance, a notion that takes a central position in IR evaluation. The purpose of automatic retrieval strategy is to retrieve all the relevant documents at the same time retrieving as few of the non-relevant as possible (Rijsbergen 1979).

All systems designed to help humans perform tasks should be evaluated; but IR systems exhibit in extreme form some of the characteristics that make this process necessary. In particular, the task(s) to be performed by an IR system are not in general very well defined — certainly not to the point where it is obvious whether or not the computer has succeeded. On top of that, there any many plausible possibilities, and many plausible arguments in favour of one or other possibility, but absolutely no guarantee that any such argument is water-tight (Robertson 2000).

Work on MIR systems has, in general, focused more on development rather than evaluation (Downie 1999). MIR evaluation test-beds and performance measures similar to the Cranfield model in text retrieval (Cleverdon 1967) — or its successor, the TREC (Text REtrieval Conference) model (http://trec.nist.gov) — are in their early development stages (Downie 2002b).

Another known problem with IR systems is that for a given query, the most highly relevant document is not retrieved with rank number one[2]. One of the possible reasons for this is query precision, where queries are generated from erroneous inputs. Of the various query-by-melody interfaces for MIR systems, one that has been gaining popularity is QBH (query by humming),

[2]The list of documents retrieved for a given query are often sorted based on a score giving rise to a ranked list.

where users query a MIR system by humming. For example, to retrieve all versions of a piece from a digital musical collection, a user could hum the tune instead of typing the title or the musical contour (the directions of adjacent pitches) of the theme. This can be said to be appealing to a large number of users, whether musically literate or not. A number of studies on QBH studies have addressed error models (McNab et al 1996; Haus and Pollastri 2001; Kosugi et al 2000) on humming errors. What is required by users is that MIR systems (more precisely QBH systems in this case) should work for everybody — perfect singers or not (Prechelt and Typke 2001). Fault-tolerant or error-tolerant QBH systems are necessary to provide for the large number of unprofessional singers who need to use the system.

Music data can be encoded digitally in one of several digital music formats. These formats in general are categorised as a) highly structured formats such as the Humdrum (Huron 1997) and GUIDO (Hoos et al 1998) where every piece of musical information on a piece of musical score is encoded, b) semi-structured formats such as MIDI (Musical Instrument Digital Interface) (Selfridge-Field 1997) in which the digital sound event information is encoded and c) highly un-structured raw audio that encode samples of sound energy level. Music description is multi-dimensional where musical sounds are commonly described by its pitch, duration, timbre, dynamics, moods and styles. In the MIR domain, music information has been conceived as consisting of seven facets (Downie 1999) — pitch, temporal, harmonic, timbral, editorial, textual and bibliographic. Stepping out further into psychoacoustics, a performance of music is said to contain the following seven perceptual attributes: pitch, rhythm, tempo, contour, timbre, loudness, and spatial location (Levitin 2001). In treating music as objects for content-based retrieval, where music features are extracted from raw data of music objects for indexing or further processing, Hsu et al (1998) classified music features based on the characteristics of music into three categories. These are a) static music information that refers to the intrinsic music characters of music objects such as key, beat and tempo, b) acoustical features that include loudness, pitch, duration, bandwidth and brightness, which can be automatically computed from the raw data of music objects and and c) thematic features such as themes, melodies, rhythms and chords derived from staff[3] information of a music object. Whatever multitudes or definitions to music dimensionality that there may be, with

[3]Lines on a musical sheet in which musical information is notated, analogous to ruled lines of a sheet of paper for writing text.

the current available technologies, it is not possible for all dimensions of music information to be extracted or inferred from any one particular format. The technologies required for extracting or inferring music information from these various formats can be very diverse, and this in general has currently resulted in two main streams to MIR research and development: sample-based and structured – using audio and symbolic music data respectively (Lu 1999).

With audio data, speech recognition tools adapted or enhanced for music data, signal processing and audio analysis algorithms that enable feature extraction and classification, similarity computations based on acoustic features, are amongst the underlying technologies for the development of MIR systems in this stream. MIR research prototypes that have been developed include systems that enable instrumental classification (Wieczorkowska 2000), genre classification (Pye 2000; Tzanetakis et al 2001), melody spotting (Durey and Clements 2001), query by audio example (Wold et al 1996; Foote 1999; Tzanetakis et al 2001; Pickens et al 2002). With complex music audio data, advancements in transcription technologies for the extraction of pitch and rhythm information would enable approaches very specific to the domain of symbolic data such as string-matching and text retrieval to be applied to audio data. This might be also the case with yet another class of music data — images, where OMR (Optical Music Recognition) techniques can be applied for the extraction pitch and rhythm data from sheet music (MacMillan et al 2001; Bainbridge et al 1999). With the pitch and rhythm dimensions quite easily obtainable from structured music data, music has commonly been represented as text strings. Using these, string matching (Crochemore et al 2001; Mongeau and Sankoff 1990; Lemström 2000), n-gram matching (Doraisamy and Rüger 2001; Downie 1999; Tseng 1999; Pickens 2000; Uitdenbogerd and Zobel 1999), IR models and edit distance measures are amongst the main approaches to MIR system development using symbolic data. This thesis focuses on the use of n-grams with music retrieval that has been extensively studied with monophonic[4] music data. With large collections of music available in the polyphonic form, the use of n-grams with polyphonic music retrieval will be investigated.

[4]where a single musical note is sounded at one time, as opposed to polyphonic music, where several notes may be sounded simultaneously at any one point in time.

1.2 *N*-grams and Music Retrieval

N-grams have been widely used in text retrieval, where the sequence of symbols is divided into overlapping constant-length subsequences. For example, the word 'music' comprises the bigrams mu, us, si and ic. A character string formed from n adjacent characters within text is called an n-gram (Heaps 1978). *N*-gramming has recently been adopted as an approach for indexing sound-related data (Meadow et al 2000). An experimental system by Downie (1999) using a database of folksongs allowed indexing of the entire musical work. With this approach to full music indexing of monophonic data, each folksong of the database was converted into a sequence of pitches (e.g. MIDI semitone numbers) ignoring the individual duration. Using a gliding window, this sequence was fragmented into overlapping length-n subsections. These n-grams were then encoded as 'text' words or 'musical words', a term coined by Downie (1999). The term 'musical words' will be used throughout this thesis. This is basically a string of characters with no semantic content. As a consequence, each folksong could be represented as a 'text document', and regular text search engines can be used. The length of the n-gram and coding details of a note can vary.

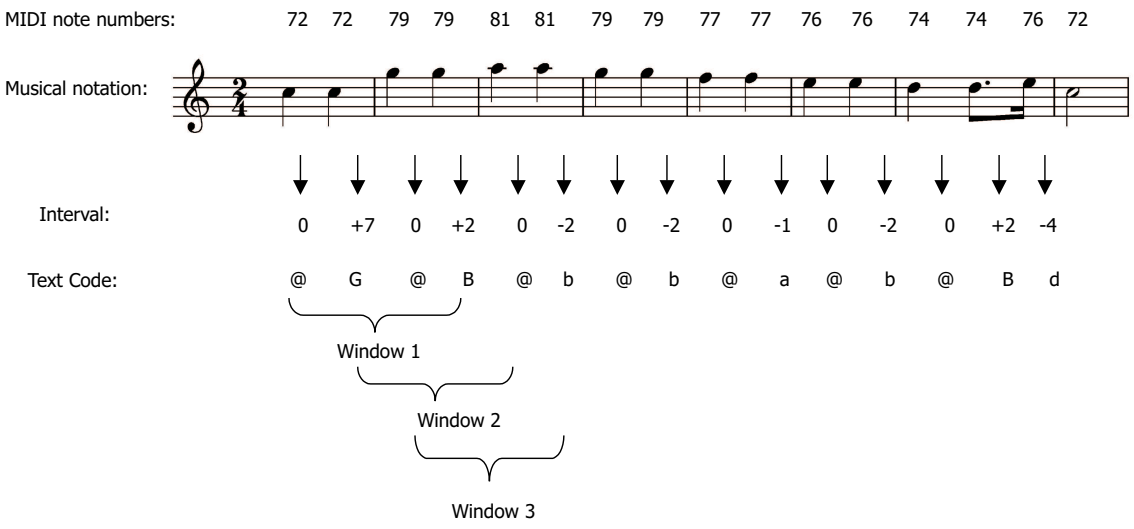


Figure 1.2: Monophonic n-gramming

From the example in Figure 1.2, the melodic string obtained using interval-only representation is 0 +7 0 +2 0 -2 0 -2 0 -1 0 -2 0 +2 -4. This is then encoded using text letters. Using one of the encoding methods outlined in Downie (1999), the first three musical words gener-

ated using 4-gram windows are: @G@B, G@B@ and @B@b. N-grams would be generated to the end of the sequence with this gliding window approach. These musical words can then be indexed using text retrieval methods.

The summary of these steps are as follows:

- code semitone differences or intervals
- assign letters to differences
- create n-grams by a gliding window
- index and query with a text search engine

1.3 N-grams and Polyphonic Music Retrieval

The n-gram approach to music retrieval has been proven successful (Downie 1999). This study continues to investigate its use with music retrieval, taking it forward into its next phase — with polyphonic data. The following subsections elaborate this research focus.

1.3.1 Significance of Study

The use of n-grams with music retrieval has largely been confined to monophonic data. A number of studies have used this approach with polyphonic data by preprocessing the collection into a collection of monophonic sequences. This approach, however, could result in a loss of information.

Full-music indexing and text retrieval systems have been extremely successful for the retrieval process from large text collections and the exploitation of this technology for music retrieval has been proven feasible. Formulating an approach for full-music indexing of polyphonic music is most certainly needed to adapt this use of text retrieval systems for music retrieval. An approach to construct n-grams from polyphonic music data using the pitch and rhythm dimensions of music is introduced. This is an extension to previous work of constructing n-grams using pitch interval information.

A data-driven approach to encoding musical n-grams into musical words is adopted whereby the encoding mechanism reflects the pitch and rhythm information of the dataset. This would be necessary in order to use existing text search engines for music retrieval.

With a large number of possible intervals and ratios, and a limited number of text characters, the use of interval classes (a range of interval values form an interval class) and ratio bins (a range of rhythmic ratios form a bin) had to be investigated. Intervals are pitch differences between two adjacent pitches and ratios refers to ratios of onset time differences between two adjacent pairs pf pitch events. An analysis of the data was performed, and a function for encoding intervals to text letters was formulated as well as one for distributing rhythmic ratios into bins. The coarseness at which pitch and rhythm information is best encoded was investigated.

Various n-gramming strategies to overcome problems identified in the use of n-grams on a polyphonic music collection were proposed. Methods to improve the retrieval precision were investigated and a new proximity based operator and scoring function for ranked retrieval of musical documents was proposed.

Due to a lack of a standardised test-bed for MIR evaluation a small-scale test collection was developed with around 10,000 polyphonic MIDI pieces. This was used to evaluate the proposed approach for the construction of n-grams for this study. This test collection was also used as a case study in one of the several workshops and meetings that are currently ongoing with the aim of developing large-scale standardised test-beds for MIR system evaluation (Doraisamy and Rüger 2003a).

A survey of error models was performed for fault-tolerance studies to be included in our experimental framework. The feasibility of this n-gram approach to polyphonic music retrieval is statistically shown for both monophonic and polyphonic queries.

1.3.2 Aims and Objectives

The main aim of our study is to propose and evaluate an approach towards the use of n-grams for the development of polyphonic music retrieval systems for large music collections.

The objectives include:

- Outline innovative methods for n-grams construction from polyphonic music data
- Outline suitable encoding mechanisms for musical words generation
- Investigate full-music indexing with various IR models for polyphonic MIR

- Investigate methods to improve the retrieval precision and fault-tolerance of indexing methods for musical text documents
- Research MIR as an experimental science based on principles of IR

1.3.3 Scope of Study

In this section we define the scope of our study within the general IR model discussed in Section 1.1.

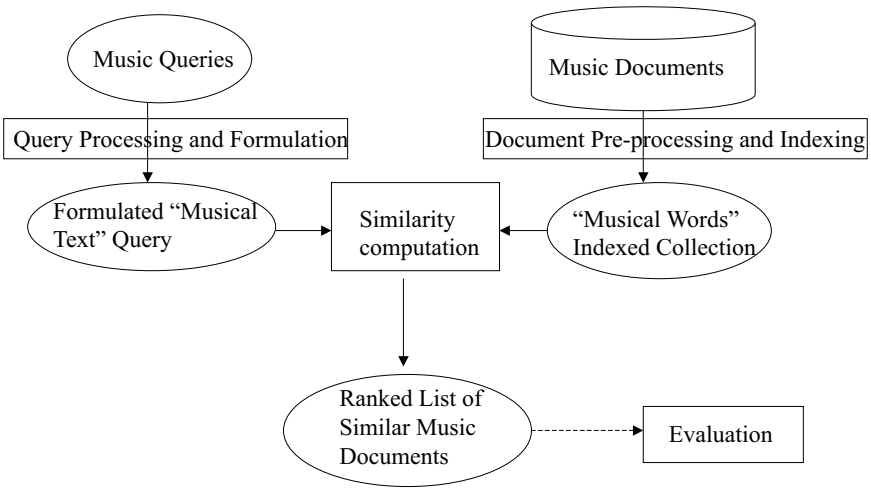


Figure 1.3: MIR process

The outline of the scope for this study follows; it is based on the MIR process shown in Figure 1.3.

- Music Documents
  - Polyphonic MIDI performances that are largely classical.
- Music Queries
  - Information need would be to retrieve all similar pieces given a musical query excerpt.
  - Two types of music queries would be investigated:
    1. Polyphonic Queries
       * Polyphonic MIDI performances as query-by-example (QBE)

2. Monophonic Queries
    * Monophonic queries as QBM (which can also be said to be one form of QBE comprising a single monophonic melody line) with input interfaces that include QBH, a graphical keyboard and text boxes. QBE would refer to polyphonic queries and QBM to monophonic queries within the context of this thesis.

- Document pre-processing and indexing
    - To propose methods to construct n-grams from polyphonic music data using the pitch and rhythm dimensions.
    - Research-based text search engines would be adopted for indexing and retrieval.
- Query processing and formulation
    - Queries and documents would be processed similarly.
    - Query formulation methods used in IR would be adopted.
- Similarity Computation
    - Music similarity assumptions would be outlined to enable evaluation.
    - Ranking approaches of various IR models would be adopted and also investigated towards the proposal of a new ranking function.
- Evaluation
    - A small-scale test collection developed using around 10,000 polyphonic music performance files.
    - A survey of error models would be performed for the fault-tolerance of indexing and retrieval proposals.
    - Text IR evaluation metrics to be used for performance evaluation.

1.4 Summary — Thesis Outline

- Chapter 1: Introduction

  The thesis begins with an introduction to the research field of MIR. The discussion provides an overview of the wide scope of this field and the main challenges that MIR researchers and developers face. MIR is discussed from the perspective of the text IR field. The use of n-grams with music retrieval is presented next. The focus of this thesis is discussed in three sections: the significance of the task undertaken, the research aims and objectives, and the scope of study.

- Chapter 2: Literature Review and Background

  This review focuses in particular on work done based on the use of symbolic data. The discussion is presented in six sections: a) musical sequence analysis, b) n-grams, c) text IR, d) the rhythm dimension of music, e) formats, and f) MIR test collections. Apart from being able to draw the rationale and research implications from the review, this chapter's aim was also provide background material that may be needed in understanding of this thesis.

- Chapter 3: Indexing

  The approach for full-music indexing of polyphonic music data with the n-gram method is outlined in this chapter. The pattern extraction method for the construction of n-grams from polyphonic music data and the data analysis for encoding musical n-grams are presented. Strategies to improve the effectiveness for indexing and retrieval based on the various problems identified are outlined. Position indexing adjacent and concurrent musical words using a 'polyphonic musical word indexer' and a scoring function for musical words is proposed.

- Chapter 4: Methodology

  This chapter presents the experimental framework for evaluating the effectiveness of the proposed approach. With no standardised test-bed available, the methodology of this study includes the development of a test collection. The retrieval performances of monophonic and polyphonic queries made on a polyphonic database are investigated.

In particular, the focus for monophonic queries is on query-by-humming and for polyphonic queries on query-by-example. The feasibility in utilising position information of polyphonic musical words is investigated using various proximity-based and structured query operators available with text retrieval systems.

- Chapter 5: Evaluation Results

  The results are presented and these are discussed based on the four stages of the experimental framework: a) preliminary investigation, b) fault-tolerance and comparative study, c) robustness and path selection and d) proximity analysis. A summary of findings based on these four stages of experiments is presented.

- Chapter 6: Conclusions and Discussions

  This last chapter concludes the study with a discussion of the findings and contributions, limitations and suggestions for future work. This includes the system design for the development of a polyphonic music retrieval system with an interface for music friendly inputs. The early prototype developed is discussed. The thesis ends with an additional section of proposals emphasizing the need for TREC-like collaboration for the development of large-scale standardised test-beds for MIR evaluation.

Chapter 2

Literature Review and Background

This chapter presents a review of work done in the area of MIR, in particular focusing on studies based on symbolic music data. Firstly, a background survey of pattern induction and matching methods that addresses the problem of ambiguity in music similarity is presented. This is followed by a review of studies that have adapted the use of n-grams and text retrieval approaches towards the development of MIR systems. Apart from the pitch dimension of music, which has been extensively used for music retrieval, several studies have focused on the use of the rhythm dimension as well, and a review of these studies is presented next. After this, the background to digital music formats is discussed briefly. Lastly, with the lack of standardised large-scale test collections for MIR evaluation, a review of small-scale test collections developed by individual MIR researchers is presented. The chapter concludes with the rationale of adopting the n-gram approach towards the development of a polyphonic music retrieval system and it's research implications.

2.1 Musical Sequence Analysis

Analysis of musical sequences to either identify perceptually significant musical patterns, such as themes and motives, or search for a given musical pattern in a musical sequence, involves human musical perception and knowledge. In general, composers repeat the theme(s) of their composition several times throughout the piece. This repetition may not be exact but can be varied (i.e., transposed, augmented, ornamented, etc.). Automated extraction and matching of musical patterns is an underlying technology for a large number of MIR applications —

such as forms analysis, copyright problems, ethno-musicology, theme indexing and query-by-melody systems (Doraisamy 1995).

It is often hypothesised that a musical surface may be seen as a string of musical entities such as notes, chords, etc. on which pattern recognition or induction techniques can be applied. Pattern induction refers to techniques that enable the extraction of useful patterns from a string, whereas pattern recognition refers to techniques that find all instances of a predefined pattern in a given string. (Cambouropoulos et al 1999).

The next two subsections presents a review of pattern recognition and induction algorithms for musical sequences.

2.1.1 Musical Sequence Matching

Several musical sequence matching algorithms have been proposed to solve the pattern recognition problem for musical sequences. Most MIR studies using symbolic data have focused on retrieving from a collection of monophonic sequences by sequential search (also known as on-line retrieval), based on either exact or approximate matching approaches (McNab et al 1997; Kornstadt 1998; Hewlett 2001). Sequences in these collections are typically complete music pieces or perceptually significant musical patterns. A summary of matching techniques that have been adapted for various MIR studies is available in Cambouropoulos et al (1999) and Lemström (2000), these are discussed in this section.

String searching finds all occurrences of a pattern[1] in a text. Pitch and rhythm information can easily be obtained by simple preprocessing of symbolic music data, musical sequences containing pitch and/or rhythm information can then be converted to text strings and string matching methods applied. Sequential searching algorithms are useful for searching texts when no data structures (such as indices) has been built over the text. Exact string matching algorithms include Brute Force, Knuth-Morris-Pratt, Boyer-Moore and Shift-or (based on bit parallelism); solutions to approximate matching include dynamic programming, automaton, bit-parallelism and filtering (Baeza-Yates and Ribeiro-Neto 1999).

Some of the relatively early MIR studies that have used exact matching approaches include

[1]Basically a pattern matching algorithm uses a window of a size equal to the length of the pattern. It first aligns the left ends of the window and text. Then it checks if the pattern occurs in the window and shifts the window to the right. It repeats the same procedure again until the right end of the window goes beyond the right end of the text (Crochemore et al 2001).

Dillon and Hunter (1982) and Stech (1981). Stech developed a programme to look into ways of locating different patterns that reoccur in a melodic line. To locate the patterns, the user had to specify the test indicating the type of melodic analysis required such as original, retrograde (backwards), retrograde-inversion (upside down and backwards), etc.. The melodic line given was broken down into smaller sequences (based on the pattern length specified by the user) and for each sequence, the test was applied to locate repetitions. With the study by Dillon and Hunter (1982), a general retrieval system was modelled to retrieve variants from a database of folksongs. The melody whose variants are being sought and the type of variant to be retrieved is defined by the user. Exact matching was also used in developing a retrieval system called the Themefinder (Kornstadt 1998). This was based on a collection of around 2000 monophonic representations of themes from a repertory of classical, renaissance and folksong. The themes contained in the database were encoded in Humdrum (Huron 1997) format, a highly-structured music format. Music formats are discussed later in Section 2.5. Using Humdrum tools, each file was converted into the several representations — Interval, Scale Degree, Gross Contour and Refined Contour — providing users with common text-based input options for musical sequences. With the problem of retrieval speed using sequential search, this preprocessing also enabled queries from the Web interface to be swiftly matched against the representations without the invocation of various Humdrum commands. Another user option available, that could also possibly reduce search time, was to specify the location to be searched. Whole sequences need not be searched if themes have to be located only at the beginning of the sequence.

To address the complexities of defining similarity between two musical sequences (as to how can two musical sequences be considered similar), several approximate matching algorithms have been adapted from text string matching algorithms for music retrieval (Mongeau and Sankoff 1990; Crochemore et al 2001; Lemström 2000; Liu et al 1999a). One widely used approach has been dynamic programming (DP) where the optimal series of transformation required to transform one sequence to another is obtained. A similarity measure and an optimal alignment between the sequences are calculated (Kruskal 1999). This was adopted for music matching by Mongeau and Sankoff (1990) where similarity measures were calculated using weights based on contributions from the pitch and duration dimensions of each note in

the musical sequences. In addition to the insertion, deletion and substitution edit operations[2] for string matching algorithms, two additional edit operations for musical sequences were introduced — consolidation and fragmentation. These involve matching a single note to repeated short notes with only a small penalty score. DP was used in retrieving from large databases in the studies by McNab et al (1997) using 9400 international folk tunes and Sonoda and Muraoka (2000) with approximately 10,000 melodic sequences. Again, retrieval speed was a problem and a state matching algorithm by Wu and Manber (1992) was then used by McNab et al (1997). However, it was reported that it did not discriminate as well as dynamic programming (McNab et al 1997). Sonoda and Muraoka (2000) proposed a short DP approach to overcome the speed problem; edit distances between shorter sequences were obtained. Indexing and retrieval was then performed based on these.

The melodic matching techniques study by Uitdenbogerd and Zobel (1999), using a collection of 10,466 polyphonic MIDI pieces, was based on a three-stage framework: melody extraction, standardisation and similarity function. Melody extraction methods were investigated to obtain monophonic melodic lines from a polyphonic collection. Standardisation involved techniques to convert melodic lines to a structure amenable to sequence matching. The aim of the similarity function section was to investigate algorithms for matching a musical query sequence against the collection of monophonic sequences extracted. The algorithms investigated were: dynamic programming, longest common sub-string and n-gram counts. With the n-gram approach, two n-gram measures were used. The first, the number common between the two strings and the second based on the Ukkonen measure (Ukkonen 1992). For obtaining the Ukkonen measure, the number of occurrences of each unique n-gram in a query and a document is obtained. The difference between the n-gram count is obtained. The Ukkonen measure is the sum of all these differences. Although in general DP performed well, the results analysis included a discussion reasoning the poor performance of the n-gram approach. These are discussed in Section 2.2.

An in-depth study on string matching techniques for music retrieval was performed by Lemström (2000). An algorithm for sequential searching proposed by Myers (1998) was adopted for the development of their MIR system, SEMEX (Search Engine for MElody EX-

[2]A very general model for similarity between two sequences is the Levenshtein distance, or simply edit distance. The edit distance between two strings is the minimum number of character insertions, deletions and replacements needed to make them equal (Baeza-Yates and Ribeiro-Neto 1999).

cerpts). The algorithm is based on a a fast bit-parallel implementation of dynamic programming. SEMEX's implementation included both the standard dynamic programming and Ukkonen's cutoff algorithm, as explained in the previous paragraph, thus enabling comparison. To retrieve with monophonic queries from a polyphonic database, an algorithm called Mono-Poly was proposed, where distributed occurrences of the query could be found. This included a filtering phase that identified candidate locations of the query pattern in a polyphonic piece. Although the filtering phase was fast, finding proper occurrences of the query pattern involved using a slower checking algorithm named Algorithm C.

Several approximate string matching approaches were also investigated by Crochemore et al (2001) for musical sequences. The Tuned-Boyer-Moore, Skip-Search and Maximal Shift string matching algorithms were adopted for their study. Further reading on these string matching algorithms are also available in Sunday (1990). The research group at the National Tsing Hua University (NTHU), Taiwan proposed approaches for approximate string matching, pattern extraction and tree indexing for their music retrieval project called *Muse* (Chou et al 1996; Hsu et al 1998; Chen and Chen 1998; Liu et al 1999a; Liu et al 1999b; Chen et al 2000; Lee and Chen 2000). The approximate string matching algorithm proposed by Liu et al (1999a) was based on the edit distance measure. A similar approach to Mongeau and Sankoff (1990) of assigning varying weights[3] based on tonal differences between two notes was adopted. Implementation details using a linked list data structure were outlined. Although the approach proposed was said to be generic to pitch, rhythm and chord strings, the tests performed only included pitch strings. Melody, rhythm and chord strings had been classified as thematic feature strings, amenable to the string matching approach proposed.

The use of multi-dimensional searching routines had been proposed for MIR as an alternative to string matching techniques (Reiss et al 2001). Based on a comparative study of several routines, the KD-Tree (K-dimensional binary search tree) for multidimensional nearest neighbour searching was suggested. This would have to be evaluated using music data as this approach is memory intensive and careful building of the binary search tree is needed. A very recent development with polyphonic music retrieval based on symbolic data is a statistical approach to harmonic modelling (Pickens and Crawford 2002). A music score is preprocessed

[3]Unlike differences between two alphabet characters, such as C and E, that would be considered dissimilar with text retrieval, a similarity measure between 0 and 1 was assigned based on tonal similarity within degrees of a Western Musical scale.

to probabilistically analyse its underlying harmonic structure. Monophonic queries could be posed on a polyphonic collection and it successfully retrieved variations of a piece. However, retrieval based on harmonic properties excludes a large amount of music data such as atonal music, contemporary or non-Western music, or even computer music that may not contain clear harmonic properties.

2.1.2 Musical Pattern Induction

A number of musical pattern extraction methods from both monophonic and polyphonic music pieces (Uitdenbogerd and Zobel 1999; Meredith et al 2002; Dovey 2001; Hsu et al 1998; Liu et al 1999b; Tseng 1999; Smith and Medina 2001; Meek and Birmingham 2001; Iliopoulos et al 2000) have been proposed in recent years. Most repetitions in music are not interesting and identifying perceptually significant repetitions is a challenge. Many examples and difficulties in identifying such patterns in polyphonic music have been illustrated in Uitdenbogerd and Zobel (1999) and Meredith et al (2002). The use of significant patterns to represent a music document could reduce search time, in particular if sequential search and approximate pattern matching methods are used for retrieval.

With most musical sequence matching and retrieval algorithms developed being applicable only to monophonic sequences, pattern extraction has also been useful in the filtering phase of polyphonic music retrieval systems development cycle (Uitdenbogerd and Zobel 1999; Tseng 1999). Polyphonic collections are preprocessed and simplified into monophonic collections. Uitdenbogerd and Zobel (1999) investigated heuristics capable of capturing the (monophonic) musical line that best represents a passage of polyphonic music. In their experiments with human listeners, a heuristic that always chooses the highest note of a chord performed the best. Of the four melody extraction methods based on the heuristic that were proposed and tested, the technique referred to as all-mono performed best. In this method, all channels were combined and the highest note from all simultaneous note events were kept. A fast algorithm that had been proposed for multilingual keyword (or key-phrase) extraction in text was modified for extracting repeating patterns in musical sequences by Tseng (1999). With text, key-phrases of any length could be identified; when used in character level, single words or word stems can be identified as well as multiple word phrases. In extracting patterns from polyphonic music, Tseng (1999) assumed each track to be separate musical sequences and

repeating patterns were identified from each of these for indexing purposes.

Algorithms proposed by the NTHU group in extracting significant patterns from monophonic sequences included an approach by Hsu et al (1998) using correlative matrix and Liu et al (1999b) based on the tree data structure called the RP-tree (RP stands for repeating pattern). Candidate repeating patterns strings form nodes, as opposed to characters from strings that are typically used in building search trees. The RP-tree approach was shown to be more efficient than using correlative matrix or suffix trees[4]. As for extracting polyphonic patterns from polyphonic sources, an algorithm has very recently been proposed by Meredith et al (2002). A geometric approach as opposed to string-based approaches is taken in which the music is represented as a multidimensional dataset[5]. Polyphonic music data is converted to a dataset of coordinates and translation vectors for all the points are stored in a vector table. For isolating theme-like and motive-like patterns in a passage, heuristics in reducing these points were based on regional proximity. However this enables only exact matches to be identified and not approximate. With music retrieval, similar pieces are not always matched exactly to be considered similar (a sequence that is varied a little musically could be considered similar). This may have to be addressed through approximate matching methods and similarity measures with cutoff thresholds.

2.2 N-grams

This section first briefly discusses the conventional use of n-grams, i.e., with text retrieval. Next, with more recent technologies, a discussion on their use with music retrieval is presented.

2.2.1 Textual String N-grams

N-grams have a number of applications, including analysis of printed English text, spell checking, cryptography, and data compression (Comlekoglu 1990). The first use of n-grams were by cryptographers in World War II (Kowalski 2000). Shannon (1948) investigated modelling printed English text as a discrete source of information and used conditional probabilities of

[4] A data structure originally developed for substring matching that can also be used for finding repeated patterns (Hsu et al 1998).

[5] A multidimensional dataset is a finite set of position vectors or data-points in a Euclidean space with a finite number of dimensions (Meredith et al 2002).

overlapping n-grams in the language to do a series of approximations to English (Shannon 1948), and later predicted the entropy of printed English text using overlapping n-grams (Shannon 1951). Information theory, as developed in mathematical terms by Shannon, is concerned with information content in terms of the amount of information required for identification of symbols or words rather than in terms of the knowledge communicated by them (Heaps 1978). Informetric[6] analysis of n-grams in English has been performed by a number of researchers and a discussion of these is available in Heaps (1978).

N-grams are useful as index terms, studies that have investigated this include Comlekoglu (1990) and Adams (1991). With n-gram indexing, a whole word is not indexed but substrings from a word are. The robustness of the n-gram approach is discussed in more detail in Chapter 3. An in-depth discussion on the use of n-grams as an indexing data structure for the development of IR systems is available in (Kowalski 2000). N-grams are viewed as a special technique for conflation (stemming) and as a unique data structure in information systems. In general with stemming, the stem (when prefixes and suffixes are removed, such as the word 'go' from 'going') of a word is determined that represents its semantic meaning. N-grams do not involve semantics. It is algorithmically based upon a fixed number of characters. The choice of the fixed length word fragment size has been studied extensively, particularly the trade off between information and the size of data structure (Heaps 1978; Comlekoglu 1990). The advantage of n-grams is that they place a finite limit on the number of index terms or searchable tokens, given by Kowalski (2000).

$$\text{MaxSeq}_n = (\lambda)^n \tag{2.1}$$

The maximum number of unique n-grams that can be generated, MaxSeq, can be calculated as a function of n which is the length of the n-grams, and λ which is the number of processable symbols from the alphabets.

N-gram indexing is a robust indexing approach that enables queries with partial terms (Witten et al 1999) or queries that are erroneous. There are also no constraints on the text to be indexed and it has proved useful in indexing OCR-degraded-text (Harding et al 1997)

[6]Informetrics is the study of the quantitative aspects of information (Tague-Sutcliffe 1992). It is a valuable tool with which to design and maintain IR systems where quantitative properties of information could include the distribution of index terms and the database growth rate (Wolfram 1992).

or documents in multiple languages. With high probabilities of errors in music queries, the use of n-grams of musical sequences has been studied towards addressing fault tolerance in music retrieval (Downie 1999; Doraisamy and Rüger 2003c).

2.2.2 Musical Sequence N-grams

N-grams can form discrete units of melodic information much in the same manner as words are discrete units of language (Downie 1999). N-grams are n consecutive characters in a text with no assumptions imposed on the text (Tseng 1999). Conversion of musical sequences into text strings enables construction of 'musical n-grams' that represent musical information. In this section studies that have investigated the use of n-grams with music retrieval (Downie 1999; Uitdenbogerd and Zobel 1999; Tseng 1999; Pickens 2000) where n-grams have been constructed from monophonic sequences with pitch only information are reviewed. In adopting text retrieval methods, varied approaches have been taken in encoding the musical n-grams amenable to the use of text retrieval systems.

Monophonic sequences extracted from the polyphonic test collection in the study by Uitdenbogerd and Zobel (1999) were converted to contours, interval sequences and modulo interval sequences using pitch information. N-gram matching was one of the sequence matching methods evaluated in this study, and the value n=4 based on their experience of its use in genomic and string matching was adopted. Varying n trades recall against precision[7] — high n provides closer matching but is more likely to miss variations with omitted or additional notes. More detailed discussion on standardised evaluation is presented in Section 2.6. The n-gram method showed a relatively poor performance in comparison to the other approaches and it was discussed that the n-gram count method may have been penalised because some n-grams, such as when the same note is repeated several times, are very common, thus favouring long pieces of music when queries contained these common n-grams. Shorter queries are less likely to contain the common n-grams and perform about as well as long queries. It was suggested that it may be appropriate to use n-gram frequencies within the collection to determine a contributory weight for each n-gram, just as words are differentially weighted in

[7]Standard retrieval performance evaluation measurement technique include recall (the proportion of the relevant melodies that have been retrieved) and precision (the proportion of the retrieved melodies that are relevant). These are defined later in Subsection 2.6.3.

document retrieval. IR models address this and these are discussed in Section 2.3.2.

An integrated system was developed using meta-data and content-based indexing of significant repeating patterns extracted in the study by Tseng (1999). The MIDI music format usually contain additional information to the score such as titles and composers, these were viewed as useful bibliographic metadata. For content-based indexing, the extracted patterns were indexed using n-grams. A similarity function was formulated for retrieval based on the frequencies of n-grams appearing in the query and the document. Their library's OPAC (Online Public Access Catalogue) system was used with slight change in the user interface. This integrated system would be an example of a system developed based on the FBIR (full-text bibliographic information retrieval systems) model[8], the model adopted for the MIR system development in the the study by Downie (1999). A detailed discussion of MIR systems categories is available in Downie (1999). Retrieval performance experiments included comparing the performance of indexing n-grams constructed from absolute pitch sequences and contours, and varying values n=2 and 3. In general, it was suggested that a larger value of n would be required and that contours did not discriminate well enough amongst the music documents for retrieval.

Downie (1999) investigated an approach to full-music indexing using n-grams and text retrieval systems. Monophonic sequences were converted to 'melodic strings' using interval-only representation and musical words were generated from n-grams obtained from these strings. The validity of the words were proven based on a variety of informetric analyses. The informetric analyses involved examinations in which ways the information properties of 'musical words' and 'real words' were similar or different. The validity was also based on evaluating databases developed from the musical words using simulated queries and text retrieval evaluation measures. Results of this informetric analysis performed on a collection of around 10,000 musical pieces and a detailed discussion of it is available in Downie (1999). Interval classification schemes were also investigated in encoding the musical n-grams into musical words. This was based on the notion that there exists a substantial group of interval types that occur very infrequently. It is necessary to develop a set of heuristics to apply, such that the resulting classification schemes pooled these rare events into classes that most efficiently represented the melodies, given the prior constraint that directional information

[8]FBIR systems are those which combine both full-text content and structured bibliographic metadata.

be preserved. Efficiency was defined to be the classification scheme's ability to minimise the amount of information lost through its classification process. Another study that similarly addressed interval classification schemes and fault-tolerance was the study Lemström (2000). In encoding the strings, a set of symbols had been used together with an approach named QPI (Quantised Partially overlapping Intervals) classification.

Only unigrams and bigrams were investigated by Pickens (2000) using musical sequences based on pitch intervals from the similar collection of folksongs used by McNab et al (1997) and Downie (1999). The encoding scheme was based on a simple function formulated to convert unigrams and bigrams of the interval sequences representing the direction and distance of adjacent pitches. Intervals represented by a sign and a digit were converted to a two digit unsigned number based on the analysis that there were no intervals larger than +24 or smaller than -24 in the music collection. The bigrams were converted to a four digit representation. This enabled the use of text retrieval systems for indexing whereby each of these two or four digit numbers were considered as a text term similar to words in a text to be indexed. A comparative study towards the use of the IR models and term adjacency with MIR was performed and these are discussed in the following section on Text IR.

2.3 Text IR

Text retrieval based on large databases has been studied intensively for a considerable amount of time in comparison to music retrieval. Several studies have explored some of the capabilities and limitations of current text IR as applied to the task of music retrieval (Downie 1999; Pickens 2000; Clausen et al 2000).

2.3.1 Indexing

An obvious option in searching with a basic query is to scan the text sequentially. This is time consuming, and data structures such as indices would need to be built over the text to speed up the search. The need for indices with music data was clear from comparative studies by Lemström et al (1998) and Sonoda (Sonoda and Muraoka 2000). Main indexing techniques for text include inverted files, tree indexing, and signature files (Baeza-Yates and Ribeiro-Neto 1999) of which the two that have been applied to music retrieval are discussed.

In this thesis, similar terminology as Witten et al (1999) is used. A document collection or document database can be treated as a set of separate documents, each described by a set of representative terms, or simply terms, and that the index must be capable of identifying all documents that contain combinations of specified terms or are in some other way judged to be relevant to the set of query terms. A document will thus be the unit of text that is returned in response to queries.

Inverted Document List

Inverted files[9] are a popular approach to indexing large text data collections and have more recently been adopted for the development of MIR systems. These were largely based on systems developed using monophonic sequences. One study that obtained 'words' from polyphonic music is the study by Clausen et al (2000) using a collection of around 12,000 classical pieces encoded in the MIDI format. A piece of music is represented as a set of notes where each note is described as a pair [t, p] with onset time t and its corresponding pitch event p. Each pair of onset time and pitch in a music piece was indexed as a word occurrence in a piece using inverted file index. Using the time-grid of a 16th note, Bar 1 of Figure 2.1 (a) would be represented with the data — [2,60], [4,62], [6,64], [8,65], [11,67], [11,65], [12,64]. This excerpt from J.S. Bach's *Fugue 1* and Part 1 of the Well-tempered Clavier is visualised using the given time-grid in Figure 2.1 (b). The rhythm dimension of music was addressed in this study, and this is discussed in Section 2.4.

Tree Indexing

In indexing, one model is to view the text as one long string instead of a set of words. Each position in the text corresponds to a semi-infinite string (sistring), the string that starts at that position and extends arbitrarily far to the right, or to the end of the text and queries can be based on any substring of the text (Frakes 1992). The NTHU research group have investigated this indexing approach extensively. This was seen as more suitable for searching music data than searching using keywords. Keywords would be more easily identifiable from a sequence of words compared to a sequence of musical pitches. In the study by Chou

[9] An inverted file contains, for each term, an inverted list that stores a list of pointers to all occurrences of that term in the main text (Witten et al 1999).

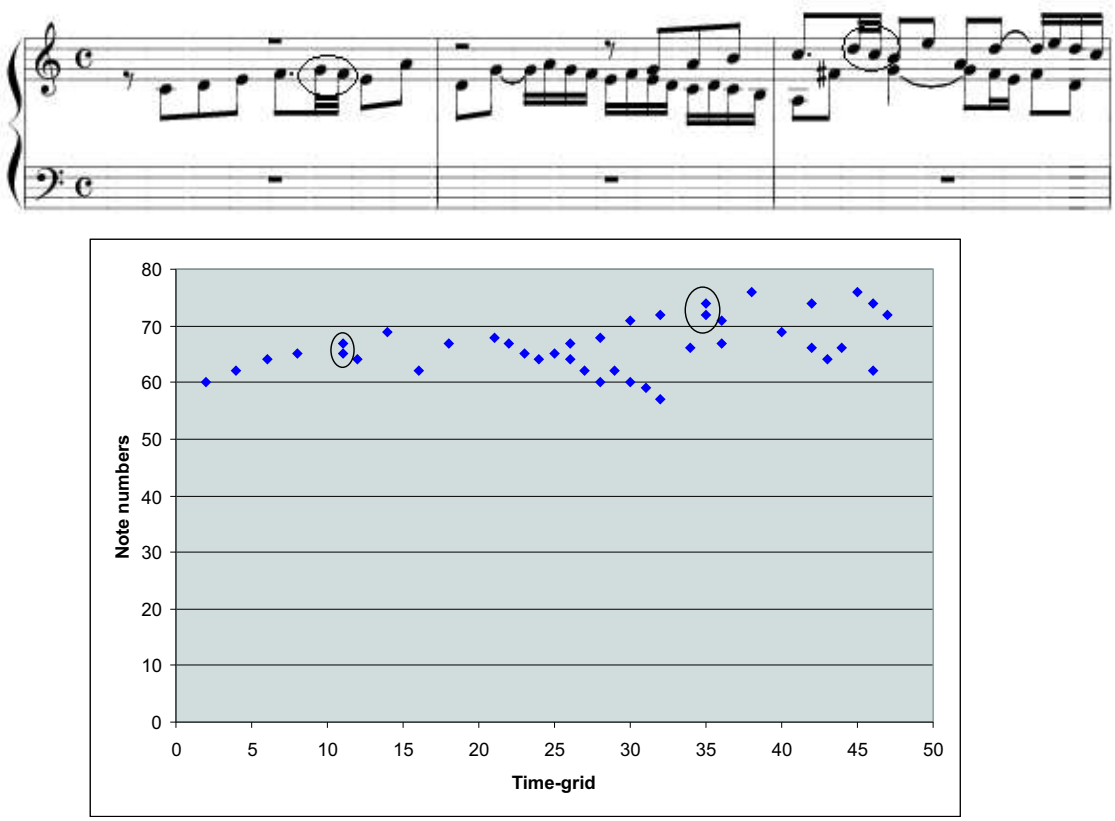


Figure 2.1: a) Excerpt from J.S. Bach's *Fugue 1* from Part 1 of the WTC and b) The polyphonic events shown on a time-line

et al (1996), Patricia (Practical Algorithm To Retrieve Information Coded in Alphanumeric) tree or PAT-tree, a trie[10] data structure was proposed as the index structure. Strings were constructed based on chords of a musical sequence. A chord-representation model and a chord decision algorithm was outlined. Chord information was used as it addressed fault-tolerance with errors that are highly likely to happen with singing inputs by singers who are not necessarily expert singers. Chord-sets were tabled and for each measure, the chord is determined using the chord decision algorithm and this sequence of chords form the string. However, the development of a chord decision algorithm based on chord progression patterns from the pop music data set might not be that easily adapted for other classes of music such as non-Western music.

The feasibility of indexing musical sequences had been investigated by Lemström et al

[10]Tries are recursive tree structures that use the digital decomposition of strings to represent a set of strings and to direct the searching. Further reading on this data structure is available in Frakes (1992).

(1998), comparing the retrieval performance of the Boyer-Moore algorithm for linear scanning of the database and a tree indexing approach using suffix-tries. The Boyer-Moore algorithm is one of the several exact string matching algorithms listed in Section 2.1.1. Results showed that indexing was needed as the database size grew larger. However, it was also discussed that in using suffix-trie for indexing, the efficiency of the indexing was also dependent to the fitting of the suffix-trie in the main memory. From the experiments, the suffix-trie structure grew rapidly and the use of a more compact but complex form of the tree, a suffix tree was proposed. The size of the tree could be restricted by reducing the depth of the tree. In a study by Gonnet et al (1991) on suffix trees, it was observed that applications slowed down as well because still only portions of the tree remained in main memory. Implementation of these algorithms on large collections of polyphonic musical sequences would certainly pose a problem.

2.3.2 IR Models

With IR systems, predicting the documents that are relevant from those that are not is usually based on a ranking algorithm. The algorithm attempts to establish a simple ordering of the documents retrieved and the first-ranked item of this ordering usually represents the document that the system has determined as having the strongest similarity with the query. The length of the returned listing is usually limited by some predetermined threshold of similarity between query and documents. A ranking algorithm operates according to various notions of relevance which yield distinct IR models, whereby the model adopted differentiates relevant and non-relevant documents. Classic IR models include the Boolean, vector-space and probabilistic models (Baeza-Yates and Ribeiro-Neto 1999).

The Boolean model is a simple retrieval where queries are specified as boolean expressions which have precise semantics. Boolean queries are keywords/query terms connected with Boolean logical operators (AND, OR, NOT). The drawback is that the retrieval is based on a binary decision, i.e., a document is either relevant or not, and not based on a grading scale (Baeza-Yates and Ribeiro-Neto 1999).

Improvements to this have been made by using information about the statistical distribution of terms, that is the frequencies with which terms occur in documents, document collections, or subsets of documents. Distinct index terms have varying relevance when used

to describe document contents and this effect is captured through the assignment of numerical weights to each index term of a document (Baeza-Yates and Ribeiro-Neto 1999). Given a set of index terms for a document, not all terms are usually equally useful in describing the document contents and deciding on the importance of a term can be difficult. However, there are properties of an index terms that are easily measured and its potential could be evaluated. For instance, a word that appears in all documents would not be as useful as an index term when compared to a word appearing in just a few documents. The latter might be useful as the number of documents that the user might be interested in is narrowed down (Baeza-Yates and Ribeiro-Neto 1999). This observation gives rise to the basic tf $\cdot$ idf[11] (Salton 1989) weight for terms.

Weights can be used with the vector-space model of an information retrieval system. A measure of similarity between a vector of term weights representing a document can be compared with a similar vector representing a query. The cosine of the angle between document and query vectors is the usual choice for similarity (the cosine rule) (Meadow et al 2000).

The inverse document frequency or the idf factor (Spärck Jones 1972) is given by

$$\text{idf} = \log \frac{N}{n_i}, \tag{2.2}$$

where n_i is the document frequency of term i, i.e., the number of documents a term i appears in and N is the total number of documents in the system. The idf component overcomes the problem brought about by common words (e.g. function words such as 'who', 'and', 'the'): A document with enough appearances of a common term would otherwise be ranked first if the query contains that term, irrespective of other words. The solution is for the term weights to be reduced for terms that appear in many documents, so that a single appearance for a word such as 'and' counts far less than a single appearance of say, 'TREC'. This can be done by weighting terms according to their idf (Witten et al 1999). This basic weighting approach has now evolved with many variants to obtaining the tf and idf values.

Text IR models have been adopted for the development of MIR systems. Downie (1999) adopted the vector space model, and Hoos et al (2001) and Pickens (2000) used the probabilistic model. The tf $\cdot$ idf ranking method, using the term and document frequencies as

[11] tf stands for term frequency, the number of times a term appears in a document; idf stands for inverse document frequency.

described above, available in the SMART (developed at Cornell University, USA) Information Retrieval System Version 11.0, was selected for use in the retrieval tests in the study by Downie (2002b). Ranking retrieval methods were selected over Boolean approaches based on the advantages that i) adjacency operations (terms/words have to appear together) or field restrictions (indexing that may be restricted to keywords), necessary in Boolean systems, are not necessary in ranking systems and ii) stop-lists (words that are deemed too common such as 'the', 'a', etc. are not indexed) are not required, nor recommended, for ranking systems. Although, stop-lists would be of little use with musical words based on semantic content, adjacency operations were investigated by Pickens (2000) and this is discussed in the next subsection.

With the probabilistic model, given a user query q and a document d_j in the collection, the probabilistic model tries to estimate the probability that the user will find the document d_j relevant. The model assumes that this probability of relevance depends on the query and the document representations only. A more in-depth description of this model is available in Baeza-Yates and Ribeiro-Neto (1999). In the comparative study by Pickens (2000), two different systems based on the probabilistic models were used for music retrieval. The first was Inquery (developed at University of Massachusetts, USA) (Callan et al 1992) based on the generalised framework of a Bayesian network. The second was language modelling, a relatively new probabilistic approach to text retrieval. Bayesian networks are directed acyclic graphs (DAGs) in which the nodes represent random variables, the arcs portray causal relationships between these variables, and the strengths of these causal influences are expressed by conditional probabilities (Baeza-Yates and Ribeiro-Neto 1999). In IR, as described by Baeza-Yates and Ribeiro-Neto (1999), the model associates random variables with the index terms, the documents and the user queries. A random variable associated with a document d_j represents the event of observing that document. The observation of the document d_j asserts a belief upon the random variables associated with its index terms. Thus, observation of a document is the cause for an increased belief in the variables associated with its index terms. Further decription of this model is available in Baeza-Yates and Ribeiro-Neto (1999). The language model adopted was based on a system developed by Ponte and Croft (1998). Language modeling combines indexing and retrieval into a single model. The language model is not only used as an index, but as a method of estimating the probability of generating

a query (Pickens 2000). Details of the use of language models in the system by Ponte and Croft (1998) are also available in Pickens (2000). This was a first attempt on using language modelling for music retrieval and the approach did not yield impressive results. However, this approach adapted for music retrieval known as harmonic modelling has shown some promising results for polyphonic music retrieval (Pickens et al 2002). Harmonic modelling has been discussed in Section 2.1.1.

Apart from the classic models, structured text retrieval models combine information on text content and document structure. Query languages to express queries through richer expressions and more precise query specification are used. In using the Boolean model, a classic IR model, for a query 'white house' to appear near the term 'president', it would be expressed as ['white house' and 'president']. A richer expression such as 'same-page(near('white house','president'))' would require a structured query language (Baeza-Yates and Ribeiro-Neto 1999). These enable combining the specification of strings (or patterns) with the specification of structural components of the document (Baeza-Yates and Ribeiro-Neto 1999), useful for proximity-based retrieval. More discussion on structured query operators and structured query retrieval is presented in Section 3.6. With MIR, the use of structured queries and musical word position with indexes addressing term adjacency have been investigated by Pickens (2000) and Doraisamy and Rüger (2003b). Term adjacency is discussed in the following subsection.

2.3.3 Term Adjacency

The inverted file structure is composed of two elements: the vocabulary and the occurrences (Witten et al 1999). The vocabulary is a set of all the different words in the text and for each such word a list of all text positions where where the word appears is stored. The granularity (the level of detail of the information stored within the index file) of the index can differ based on the resolution of term locations. Apart from just storing the document id, the location of the occurrence of a term within-document position(s) can be added to the term occurrence data. With exact positions where a word appears in a text, phrase queries can be made. For a query that consists of several words/terms, one can form a phrase of words to find documents that contain consecutive occurrences of the term sequence. For searching using phrases on the Web, one formulates the query by enclosing the terms between quotation marks. A proximity

query is a more relaxed version of the phrase query. A sequence of terms are given as a phrase together with a maximum allowed distance between them. The words and phrases may or may not be required to appear in the same order as the query. The first word's location is identified and if all terms are found within a particular distance, the term frequency is incremented for the given phrase/proximity query (Baeza-Yates and Ribeiro-Neto 1999). In general, proximity information can be quite effective at improving the precision of searches (Baeza-Yates and Ribeiro-Neto 1999). Term positions for the improvement of retrieval performances with IR systems have been extensively studied (Callan 1994; Hearst 1996; Keen 1991). With text documents, 'consecutive words' means a sequence of words according to the 'reading order' of the textual document (i.e., from the first page to the last) (Chiaramella 2000).

With polyphonic music, not only the 'listening order'/'playing order' based on the time-line has to be considered but also the concurrency of this 'order'. Terms or musical words adjacency was studied by Pickens (2000) using monophonic music data. It was said that the only sequence that has been preserved with the n-gram approach of musical word generation for indexing has been n contiguous notes.

2.3.4 Search Engines

Text search engines used for this study include MG-1.2.1 (Managing Gigabytes) and the Lemur Toolkit (2001). The latter was adopted as main tool for this study, with MG only used for preliminary investigations. Lemur Toolkit (2001) supported a larger number of IR models. Figure 2.2 shows a flowchart of a general search engine. The search for relevant documents usually involves the use of an inverted file to produce statistically ranked output. First, a query is commonly parsed using the same parser used in the index creation. Each term is checked against a stop-list for removal of common terms, and if is not common, it is passed through the stemming routine. A search is then performed for the stem against the dictionary. Obtaining weights and computing similarity would be based on the particular IR model adopted. From these scores, a ranked list of relevant documents is presented to the user (Harman 1992).

The MG search engine was developed as a collaboration project between several universities in Australia and New Zealand, including University of Waikato where Meldex (McNab et al 1997) was developed. The IR model adopted is the vector-space IR model. Details of

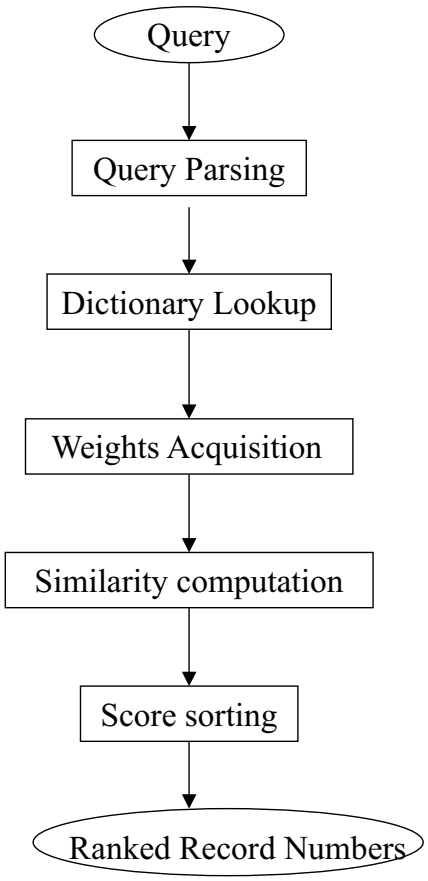


Figure 2.2: Flowchart of a general search engine

the tf · idf variants adopted for this search engine are available in Witten et al (1999).

The Lemur Toolkit for Language Modelling and Information Retrieval was developed as part of the Lemur project, a collaboration between the Centre for Intelligent Information Systems (CIIR) at the University of Massachusetts and the Language Technologies Institute (LTI) at the Carnegie Mellon University. The toolkit supports indexing of large-scale text databases and implementation of retrieval systems based on the probabilistic and vector-space IR models (Lemur Toolkit 2001).

Lemur supports two types of indexes: one storing bag-of-words representations for documents, the other storing term location information (Ogilvie and Callan 2001). As for IR models, the vector space and probabilistic models are supported. The vector space model is supported using three variants of tf weights: raw tf (within-document frequency), log(tf) (gives diminishing returns as term frequencies increase), and the Okapi BM25 tf (details

discussed below) weight.

In the Lemur toolkit, one variant of the tf $\cdot$ idf family that was implemented is based on the Okapi term frequency formula originally derived from a probabilistic model (Robertson et al 1994). The description below was adapted from Zhai (2001).

It is assumed that each document and each query are represented by a term frequency vectors:

$$\vec{d} = (x_1, x_2, \ldots, x_n) \tag{2.3}$$

$$\vec{q} = (y_1, y_2, \ldots, y_n) \tag{2.4}$$

n is the total number of terms, or the size of vocabulary and x_i, y_i are the frequency (i.e., the counts) of term t_i in d and q, respectively. Given a collection C, the well-known inverse-document-frequency (idf) of a term t is given by $\log(N/n_t)$[12], where N is the total number of documents in C and n_t is the number of documents with term t. All terms in a query or a document are weighed by the heuristic tf-idf weighting formula. That is weighted vectors for $\vec{d}$ and $\vec{q}$ are:

$$\vec{d'} = (\mathrm{tf}_d(x_1)\,\mathrm{idf}(t_1), \mathrm{tf}_d(x_2)\,\mathrm{idf}(t_2), \ldots, \mathrm{tf}_d(x_n)\,\mathrm{idf}(t_n)) \tag{2.5}$$

$$\vec{q'} = (\mathrm{tf}_q(y_1)\,\mathrm{idf}(t_1), \mathrm{tf}_q(y_2)\,\mathrm{idf}(t_2), \ldots, \mathrm{tf}_q(y_n)\,\mathrm{idf}(t_n)) \tag{2.6}$$

The document term frequency function tf_d is given by the Okapi tf formula

$$\mathrm{tf}_d(x) = \frac{k_1 x}{x + k_1(1 - b + b\frac{l_d}{l_C})} \tag{2.7}$$

with parameters k_1 and b; l_d represents the document length and l_C the average document length. The query term frequency function tf_q is defined similarly with a parameter l_Q

[12]However, via communications on the public forum with developers (Lemur Toolkit 2001), a normalised version of the idf factor is now used with the software, given by

$$\log \frac{N+1}{n_t + 0.5}$$

representing average query length.[13]

The score of document $\vec{d}$ against query $\vec{q}$ is given by

$$s(\vec{d},\vec{q}) = \sum_{i=1}^{n} \mathrm{tf}_d(x_i)\,\mathrm{tf}_q(y_i)\,\mathrm{idf}(t_i)^2 \tag{2.8}$$

There are five parameters to set: k_1 and b for the document tf function and k_1, b, l_Q for query tf. (Robertson et al 1994) suggested a default value of 1.2 for k_1 and 0.75 for b for document tf, but the change of k_1 and b may affect the performance, sometimes significantly. b is usually set to 0 and k_1 to 1,000 for query tf, which makes their query tf formula almost identical to the original BM25 query tf formula where k_3 is our k_1. BMxx is the best-match weighting function implemented in Okapi (Robertson et al 1994) where xx denotes variants, such as BM11 and BM15. Sometimes allowing the b for query to take a non-zero value is said to be beneficial. In that case, k_1 and b would be set in the same way as they are set for a document. l_Q is usually set to 3 for title queries.

The use of these search engines and values adopted for the various parameters for this study are discussed in Chapter 4. The structured retrieval model available within Lemur Toolkit (2001) is further discussed in Section 3.6.

2.4 Rhythm Dimension of Music

In addition to pitch, rhythm information can be important for music retrieval. Melodies are most identifiable by listeners when both pitch and rhythm are used, followed by pitch only and then rhythm only (Uitdenbogerd and Zobel 1999). Numerous studies have been carried out on the use of patterns generated from various combinations of the pitch and rhythm dimensions. These studies used either pitch information (Downie 1999; Blackburn and DeRoure 1998), rhythm information (Chen and Chen 1998) or both pitch and rhythm information simultaneously (Doraisamy 1995; Clausen et al 2000). The motivation discussion for the study by Clausen et al (2000) included that in a more general context of polyphonic music, one is forced to consider pitch and rhythm information (Clausen et al 2000).

Onset times can be very irregular as these can easily vary for the same song when per-

[13] This is different from the original query tf formula implied by the BM25 retrieval function, which does not have the parameter of average query length, but the difference is insignificant for short queries.

formed by different performers or even the same performer as it is known that two performances will usually never be identical. To deal with the problem of quantisation in rhythm information representation with irregular onset times, arbitrary time resolution was avoided and a fixed time-grid was used in the study by Clausen et al (2000). Onset times were quantised to a pre-selected time-grid, e.g., a sixteenth-note resolution. Although problems of this quantising method have been discussed in detail, such as different note combinations might have the same quantised form or a quantised melody might lead to chords (more than one note sounded simultaneously), it was adopted to ease the implementation of this early study. Queries and the pieces in the database are quantised in the same way. To illustrate the problem of this quantisation approach, the example from bars 1–3 of Bach's *Fugue I*, Bk I of the Well-Tempered Clavier (WTC) where polyphony clearly begins somewhere in the middle of the excerpt as shown in Figure 2.1 (a) is used. The music information extracted based on the pitch and duration dimension from the score is visualised on a MIDI note number versus the quantised time-grid of a 16th note graph as shown in Figure 2.1 (b). It can be noted that by selecting a time-grid of a 16th note (1 unit of time represents a sixteenth note duration), the two 32nd notes in bar 3 had to be encoded as a chord with the time value of a 16th note instead of two separate note events with time values of 32nd notes as circled in Figures 2.1 (a) and (b).

To retrieve songs by rhythm, Chen and Chen (1998) from NTHU proposed the use of rhythmic patterns within a measure. These were defined by them as 'mubols'. Rhythmic strings were generated based on 'mubols' and a tree-indexing approach was adopted to index these rhythmic strings. The study by Lee and Chen (2000) used absolute pitch and rhythm data to form musical strings and investigated developing multi-feature index structures based on suffix trees that incorporated both rhythm and pitch information. This was named twin suffix trees combining two separate suffix trees, one for pitch and the other for rhythm, with pointers linking nodes from both trees. These had scalability problems and other variants had been proposed to overcome this problem.

Automated rhythmic pattern induction techniques were proposed by Mont-Reynaud and Goldstein (1985) and Shmulevich et al (1999). Discussions were restricted to quantised rhythms, i.e., rhythms as notated in a score, without timing deviations due to a performance.

Automated beat tracking[14] is still in early research stages and large number of studies that have adopted the use of rhythm assume time signature knowledge. Programmes that are able to estimate the tempo and the times of musical beats in expressively performed music are still in early research stages (Dixon 2001).

2.5 Formats

There are several formats that music can digitally be encoded with. However, not all dimensions of music information can be extracted from all of these formats. This section presents a brief overview on the various formats, focusing in particular on the information representation and extraction of the pitch and rhythm dimensions of music.

2.5.1 Structured Formats

With highly-structured formats, the systematic music encoding enables computer-based symbolic representation of music information. Decoding and extracting pitch and rhythm information fundamental to the development of most MIR applications should be a relatively easy task with such highly structured musical data formats. Formats such as Nightingale (Byrd 2001), Humdrum (Huron 1997), DARMS (Digital Alternate Representation of Musical Scores), SCORE (Smith 1997), etc., enable digital representations of almost all music information that can be interpreted from a music score by a human. A comprehensive guide to musical codes for computer-based representation of musical information is available in (Selfridge-Field 1997).

MIDI (Musical Instrument Digital Interface), a semi-structured format could refer to a hardware interface, a file format, the data in a Standard MIDI file, or the instrumental simulation specifications of General MIDI (Hewlett and Selfridge-Field 1997). In this thesis, reference to it would be to the file format (MIDI files) and the data (MIDI data). This semi-structured file contains time-stamped data in a format outlined to work with MIDI hardware devices. MIDI originated as a real-time protocol to enable communication between separate hardware devices (e.g., between two electronic keyboards or between an electronic keyboard

[14]The task of beat tracking or tempo following could be described by analogy to the human activities of foot-tapping or hand-clapping in time with music, tasks of which average human listeners are capable (Dixon 2001).

and a personal computer), more specifically with the intent to make sound-wave frequency and duration-of-depression information obtained from an electronic keyboard interpretable across devices. These are given by MIDI pitch numbers and its corresponding note on and off time-stamps.

2.5.2 Unstructured Formats

More extensive technologies would be needed for extracting pitch and rhythm information from audio and image files that contain music information. This would include transcription and pitch tracking technologies for audio files, and OMR (Optical Music Recognition) with images.

Pitch trackers for monophonic files are relatively advanced compared to polyphonic transcription systems. Improvements in transcribing polyphonic audio data has been made in recent years (Martin 1996; von Schroeter 2000; Plumbley et al 2002; Bello and Sandler 2003). Extensive research in the UK is ongoing in this area at the Centre for Digital Music at Queen Mary, University of London (http://www.elec.qmul.ac.uk/digitalmusic). Automatic transcription has enabled polyphonic audio queries to be made on a collection of polyphonic pieces in the symbolic form (Pickens et al 2002) and would be important for indexing based on the pitch and rhythm dimensions. In using rhythm information based on data obtained from transcription, one observation is that onset detection is easier than the offset (von Schroeter 2000; Bello and Sandler 2003). The approach by von Schroeter (2000) detects up to about 90% of the notes in a 2-part acoustic piano signal. The excerpt in Figure 2.1 (a) was adapted in this study to illustrate a transcribed output. The transcribed output of the performance in bars 2–3 of the excerpt in Figure 2.1 (a) is shown in Figure 2.3.

Large collections of sheet music are now made available from Digital Sheet Library projects (Dunn and Mayer 1999; Olson and Downie 2003; Choudhury et al 2000) and with the use of OMR techniques, content-based retrieval would be possible (MacMillan et al 2001; Bainbridge et al 1999). The use of pitch and rhythm dimensions for indexing purposes would certainly reduce the problem of format constraints.

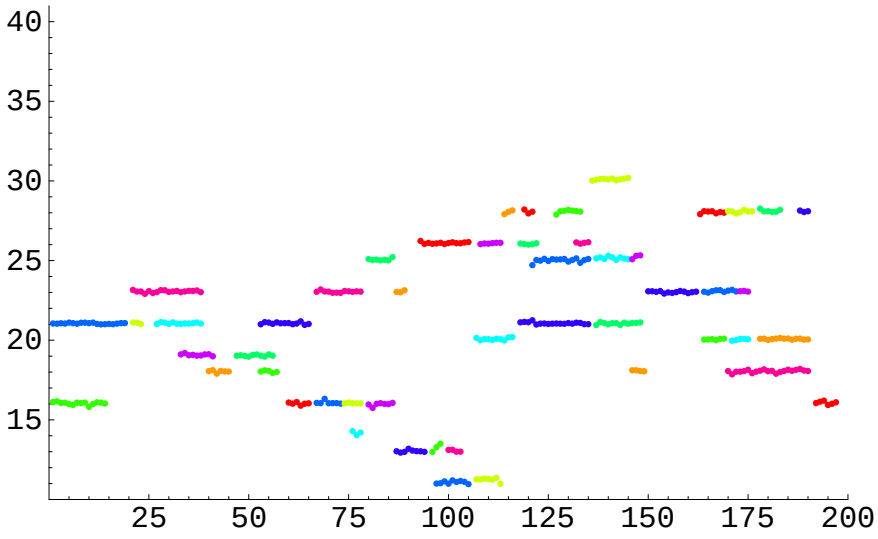


Figure 2.3: Time-pitch spectra (from von Schroeter (2000))

2.6 MIR Test Collections

The interest in MIR has grown noticeably over the past few years, and there already exist an appreciable number of MIR systems that are commercially viable and of a high degree of sophistication. However, with the lack of standardised, generally agreed-upon test collections, tasks and metrics for MIR evaluation, researchers and developers are facing difficulties in benchmarking and endorsing the performances of their systems. The need for standardised large-scale evaluation of MIR and music digital library (MDL) methodologies is currently being addressed with the recent resolution calling for the construction of the infrastructure necessary to support MIR/MDL research (Downie 2002b). The methodology of this MIR study investigating the use of n-grams for polyphonic music retrieval has been based on a small-scale test collection of around 10,000 polyphonic MIDI files developed for this study which will be discussed in Chapter 4. This development was used as a case study in Doraisamy and Rüger (2003a) emphasizing the need for TREC-like collaboration towards MIR evaluation[15]. In developing the test collection, a review of the state-of-the-art test collections developed by individual MIR researchers for the purpose of metric scientific evaluation due to this lack of a standardised test-bed was performed. This section presents this review. First,

[15]Included in the MIR/MDL Evaluation Project White Paper Collection (Downie 2003a) outlining preliminary foundations and infrastructures in establishing MIR and MDL evaluation frameworks.

the background to the TREC model adopted by the text retrieval community in developing large-scale standardised test collections is discussed. The rest of the review focuses in particular on the collection and queries used, relevance judgement assumptions and evaluation metrics that have been adopted for MIR studies.

One of the approaches proposed towards MIR evaluation is the use of test collections based on the Cranfield and TREC models (Downie 2002b; Byrd 2000) that have been used for many years in the text retrieval community. A test collection for information retrieval encompasses (i) a set of documents, (ii) a set of queries, and (iii) a set of relevance judgements, and aims to model real-world situations, so that one could expect the performance of a retrieval system on the test collection to be a good approximation of its performance in practice. Of the two series of studies conducted at Cranfield University, UK, it is the second series conducted in the 1960s that current experimental evaluation of IR systems (with the first conducted in the 1950s) (Harter and Hert 1997; Cleverdon 1967) followed. The Cranfield 2 test collection consists of 1,400 documents, mainly in the field of aerodynamics with 221 search questions (Cleverdon 1970). Further reading on the beginnings of this important development in IR evaluation is available in Cleverdon (1991). Because of computer storage and processing costs, it was not until the 1980s, however, that large-scale testing became possible. In the early 90s the Cranfield 2 paradigm was gradually replaced by TREC (Text REtrieval Conference), a major initiative in IR evaluation with an emphasis on large collection size and completeness of relevance judgements. It made possible the large-scale, robust evaluation of text retrieval methodologies and TREC has been running successfully since then (Voorhees 2002; Harter and Hert 1997). Further details on TREC are available at http://trec.nist.gov.

An important factor in the success of an evaluation is the task description and the metrics used to score the quality of a response. *Intuitively understandable metrics that map to commercially significant problems* have been described as one desirable feature of an evaluation (Voorhees 2002; Hirshman 1998). QBM, a task useful to most classes of MIR users (music librarians, disc-jockeys, music scholars, etc.), has been the research focus of a large number of MIR studies. There are various interfaces to these: QBH, text boxes for contour or absolute note letter-names input, or a graphically visualised keyboard. In this study of full-music indexing of polyphonic music, two tasks are focused on: QBM for monophonic queries and QBE for polyphonic queries (Doraisamy and Rüger 2003c).

Relevance judgements are what turn a set of documents and topics into a test collection. Deciding which documents are relevant given a particular query constitutes an important theoretical and practical challenge. The difficulties of defining relevance with music documents have been discussed in Chapter 1 and that relevance is a notion that takes a central position in IR evaluation. Relevance with music documents involves human musical perception where there are difficulties in defining when two musical documents are similar. In developing a list of relevant documents for a query, TREC uses a pooling technique, whereby diverse retrieval systems suggest documents for human evaluation. In creating judgement pools, participants would need to submit retrieval runs (one or more) for the system(s) they wish to evaluate. NIST (National Institute of Standards and Technology) decides on the maximum number of runs that a participant can submit. As described by Voorhees (2002), all runs are selected if the number submitted by a particular participant is less than the allowed maximum. When participants submit their retrieval runs to NIST, participants rank their runs in the order they prefer them to be judged. NIST merges the runs into the pools respecting this preferred ordering. For each selected run, the top X documents (usually X=100) per topic retrieved are added to the topic's pools. Documents that are not in the pool, because none of the systems ranked the document high enough, are assumed to be irrelevant to the topic. Human assessors would then be used to judge the relevance of the documents in the pool. However, the quality of the judgements created using the pooling technique should be assessed based on the completeness and the consistency of the relevance judgements (Voorhees 2002). Completeness measures the degree to which all the relevant documents for a topic have been found while consistency measures the degree to which the human assessor has marked all relevant documents as relevant and the irrelevant documents as irrelevant (Voorhees 2002).

With MIR, one of the first steps taken towards standardised music test collections was to list candidate MIR test collections. These collections are useful by themselves for a number of research projects but would be even more useful if they were accompanied by a set of well-defined queries and relevance judgements. The Uitdenbogerd (2002) collection is a notable exception as it does come with a set of (however incomplete) human relevance judgements (Byrd 2000).

In this section, a discussion on a number of MIR studies for which researchers have developed individual small-scale test collections (using between 3000 to 10,000 documents)

for evaluation (Downie 1999; Uitdenbogerd 2002; Kosugi et al 2000; Sødring and Smeaton 2002; Pickens et al 2002) is presented. Issues such as complexity of music data and queries, relevance judgements and effectiveness measures used will be focused on.

2.6.1 Collections and Queries

There are a number of additional problems that MIR researchers face when dealing with music data for computer-based MIR systems when compared to text. Music data can come as simple monophonic sequences or as polyphonic sequences. Music data is multi-dimensional with musical sounds commonly described in terms of pitch, duration, dynamics and timbre. Music data can be encoded in multiple formats: highly structured, semi-structured or highly unstructured. A few studies have included additional pre-processing modules to deal with these various aspects of music data, for the data collection and/or queries. The collection and queries used in the studies by Downie (1999) and Sødring and Smeaton (2002) were monophonic. Melody extraction algorithms were used for the studies by Uitdenbogerd (2002) and Kosugi et al (2000) to preprocess a polyphonic collection. The collection comprised the monophonic sequences obtained from the preprocessing step along with monophonic queries. The study by Pickens et al (2002) used both polyphonic queries and source collection. The collection was encoded in a highly structured format and a prototype polyphonic audio transcription system was integrated to transcribe polyphonic queries in the audio format.

Collection sizes varied between 3000 and 10,000 music files. The studies by Downie (1999) and Sødring and Smeaton (2002) used the collection by McNab et al (1996) which is also referred to as the NZDL (New Zealand Digital Library) collection in (Byrd 2000)) of about 10,000 folksongs in the monophonic format. Uitdenbogerd (2002) and Kosugi et al (2000) both used around 10,000 MIDI files where the former downloaded music of various genres from the internet and the latter obtained the collection from a company in Japan (the name of the karaoke recording company was not disclosed). Kosugi et al (2000) chose MIDI as the format for their collection as there is a large amount of MIDI available in Japan, where the popularity of karaoke ensures easy access to all the latest pop hits. Most karaoke recordings store the melody data on one MIDI channel, which makes it easy to recognise the melody (Kosugi et al 2000). The test collection developed by Pickens et al (2002) used data provided by CCARH (http://www.ccarh.org). It consists of around 3000 files of separate movements

from polyphonic full-encoded music scores by a number of classical composers (including Bach, Beethoven, Händel and Mozart). Three additional sets of variations, from which queries were extracted, were added with the final collection comprising a total of 3150 documents.

For query acquisition of the set of queries needed in developing a test collection, approaches taken were either simulation/automatic or manual. The two major categories of query construction techniques that TREC distinguishes are automatic and manual methods. An automatic method is a means of deriving a query from the topic statement without any manual intervention and a manual method is anything else. Since these methods require different amounts of (human) effort, care must be taken when comparing manual results to ensure that the runs are truly comparable (Voorhees and Harman 1999). Faced with the difficulty in obtaining real-world queries, many researchers simulate queries by extracting excerpts from pieces within the collection, and then using error models to generate erroneous queries. One such study is by Downie (1999) that consisted of two phases. In the first phase, 100 songs of a variety of musical styles were selected and queries of lengths 4, 6 and 8 were extracted from the incipits of each song. Thirty randomly selected pieces from the collection and a sub-string of length 11 from various locations in the piece were used in the second phase of the study. An error model based on the study by McNab et al (1996) was used for error simulation.

Both automatic and manual query acquisition approaches were used in the study by Uitdenbogerd (2002). Automatic queries were selected from the collection by assuming that versions of a given piece formed a set of relevant documents. Versions were detected by locating likely pieces of music via the filenames and then verifying by listening to these pieces. One of the pieces from each set of versions was randomly chosen to extract an automatic query. Manual queries were obtained by asking a musician to listen to pieces that were randomly chosen from the set of pieces with multiple versions obtained from the automatic approach, and then to generate a query melody.

In the study by Sødring and Smeaton (2002), 50 real music fragments were generated manually on a keyboard by a person with music knowledge before being transcribed into Parson's[16] notation to form the query set. 258 tunes hummed by 25 people were used as candidate queries for the study by Kosugi et al (2000). 186 tunes from these were recognised

[16] An encoding that reflects directions of melodies, either Up (U), Down (D) or Same (S).

as melodies available in the database, and hence adopted as the query set. For the study by Pickens et al (2002), the audio version of one variation was selected from each of the three sets of variations that had been added for the purpose of query extraction; this was used as the query for the QBE task. They were unable to get human performances of all these variations. Instead, queries were converted to MIDI and a high-quality piano soundfont (for the generation of synthesised sounds) was used to create an audio "performance".

Looking at a few test collections, the diversity in size, genres, formats, complexity and query acquisition approaches that have already been used in MIR studies can be seen.

2.6.2 Relevance Judgements

TREC has almost always used binary relevance judgements (a document is relevant to the topic or not). There have been studies investigating the use of multiple relevance levels (Spink and Greisdorf 2001). The most recent web track (a section in TREC to evaluate retrieval of documents on the World Wide Web) used a three point relevance scale: not relevant, relevant and highly relevant (Voorhees 2001). To overcome the difficulties in obtaining agreement on relevance, TREC uses the pooling technique to obtain a repository of candidate relevant documents and a number of human assessors to judge the relevance of these. In defining relevance for the assessors, the assessors are told: "Assume that you are writing a report on the particular topic. If the document would provide helpful information then mark the entire document relevant, otherwise mark it irrelevant. A document is to be judged relevant regardless of the number of other documents that contain the same information (Voorhees 2002)".

With the need to evaluate MIR systems, a number of relevance definitions have been assumed. With the known-item search used in the first phase of the study by Downie (1999), the document from which 100 query sequences were extracted was considered relevant and all remaining documents considered non-relevant. In the second phase, the set of relevant documents for a given query was defined as being the set of those songs in which the query's progenitor string was found intact.

In the study by Uitdenbogerd (2002), "automatic" and manual relevance judgements were used. Versions of a piece were considered to be relevant. Pieces were only considered to have distinct versions if there were obvious differences in the arrangement, such as (i) being

in a different key, (ii) using different instruments or (iii) having differences in the rhythm, dynamics, or structure. All arrangements of the piece were assumed to be relevant versions and all other pieces assumed to be irrelevant, thus giving "automatic" relevance judgements. For manual relevance judgements, six judges were asked to listen to the pieces returned by the retrieval system for relevance assessment.

The pooling approach was adopted by Sødring and Smeaton (2002) to obtain a set of relevant documents. The answer set obtained from submitting the 50 manually generated queries to their MIR system, was used as the relevant document set for the various experiments in their study. This approach was seen to be useful in generating a list of relevant documents for a set of queries without the need for human relevance judgements.

Audio queries were transcribed and submitted for retrieval for the study by Pickens et al (2002) and Kosugi et al (2000). Being one of the first studies to use an audio polyphonic query, the study by Pickens et al (2002) used one variation from each of the three query sets of variations as a query, and any variation within each corresponding set was considered relevant. With the monophonic hummed queries for the study of Kosugi et al (2000), relevant documents were identified by song names.

Despite the difficulties in defining relevance for MIR where human music perception has to be addressed, relevance had been defined within the scope of the various MIR studies, based on the need to evaluate the respective systems.

2.6.3 Evaluation Measures

Many measures of retrieval effectiveness have been defined and a number of these have been adopted for MIR studies. All the measures are based on the notion of relevance. The measures assume that, given a document collection and a query, some documents are relevant to the query and others are not. The objective of an IR system is to retrieve relevant documents and to suppress the retrieval of non-relevant documents (Harter and Hert 1997). Notable performance measures that were used for the Cranfield tests and continue to be widely used in IR are precision and recall. These were briefly discussed in Chapter 2. Formally, given a query q, a set of retrieved documents $A(q)$ and a set of relevant documents $R(q)$, then recall r and precision p are defined as

$$r = \frac{|A(q) \cap R(q)|}{|R(q)|} \quad \text{and} \quad p = \frac{|A(q) \cap R(q)|}{|A(q)|},$$

respectively.

The official TREC reports show several variants of precision and recall, such as the mean precision at various cut-off levels and a recall-precision graph. The mean average precision, explained at the end of this section, is often used as a single summary evaluation statistic (Voorhees 2000). Another measure that has been used is based on the rank of known-item search. The quality of the retrieval mechanism is judged by the reciprocal rank of the known item — e.g., if the known (and only relevant) item is retrieved at 5th rank, a quality of 0.2 would be assigned for this query. By repeating this process with many queries, a mean reciprocal rank (MRR) is obtained to assess a particular retrieval and indexing method, averaged over the number of queries. The MRR measure is between 0 and 1 where 1 indicates perfect retrieval. With the complexities of music data (Downie 2003b), other benchmarking measures beyond precision and recall have been proposed. These include evaluating based on aspects of retrieval efficiency (e.g. speed of processing), software quality metrics and human computer interaction (HCI) and user interface (UI) features (Downie 2002a).

For the first phase of the study by Downie (1999), a modified measure of precision was used. Precision was defined as: P = 1/number of song titles retrieved. Queries were extracted from 100 songs. For the purpose of the study, a non-relevant hit was any song title retrieved other than that from which the query was extracted. The second study used normalised precision and recall. The normalised precision (NPREC) and normalised recall (NREC) metrics capture how closely a ranking system performs relative to the ideal by including in the calculation, information about the ranks at which relevant documents are listed. An NPREC or NREC value of 1 indicates that the ideal has been realized while a value of 0 indicates the worst case.

The NPREC and NREC metrics are defined as (Salton 1968):

$$\mathrm{NPREC} = 1 - \frac{\sum_{m=1}^{\mathrm{REL}} \log \mathrm{rank}_m - \sum_{m=1}^{\mathrm{REL}} \log m}{\log \frac{\mathrm{N!}}{(\mathrm{N}-\mathrm{REL})!\,\mathrm{REL!}}} \tag{2.9}$$

$$\mathrm{NREC} = 1 - \frac{\sum_{m=1}^{\mathrm{REL}} \mathrm{rank}_m - \sum_{m=1}^{\mathrm{REL}} m}{\mathrm{REL}(\mathrm{N}-\mathrm{REL})} \tag{2.10}$$

Here N is the number of documents in the database, REL the number of relevant documents contained in the database, and rank_m the rank assigned to relevant document m (Downie 1999; Rijsbergen 1979).

Other standard measures adopted for MIR studies include: (i) eleven-point precision averages (recall and precision can be averaged at fixed recall levels to compute an overall eleven-point recall-precision average) (Uitdenbogerd 2002), (ii) precision at k pieces retrieved (number of relevant melodies amongst the first k retrieved) (Uitdenbogerd 2002; Sødring and Smeaton 2002), (iii) precision/recall graphs (Sødring and Smeaton 2002), (iv) mean average precision and mean precision at the top 5 retrieved documents. Average precision is computed by calculating the precision for a single query every time a relevant document is found, then averaging over all those points. This score is then averaged over all queries in the set for the mean average precision. For precision of a system at the top of the ranked list, the precision for a single query is calculated after the top k is retrieved. This would give mean precision at the top k retrieved. (Pickens et al 2002). For the study by Kosugi et al (2000), where relevance was based on the song names, the percentage of songs retrieved within a given rank number formed the basis for their evaluation.

With relevance judgements defined to a certain extent within the scope of each particular study, standard evaluation measures (modified in some cases) have already proved useful for MIR evaluation.

2.7 Summary

A review of studies that have adapted text retrieval approaches for MIR and related background technologies has been presented. As a summary to this, the chapter concludes with the rationale and research implications in adopting the n-gram approach towards the development and evaluation of a polyphonic music retrieval system.

2.7.1 Rationale

The n-gram approach to music retrieval with monophonic sequences has been extensively studied. Below is the rationale for investigating this approach towards the development of a polyphonic music retrieval system. Within the context of this thesis, this is a system that retrieves all similar music pieces from a polyphonic collection given either a monophonic or polyphonic query. The similarity assumptions are adopted from the study by Uitdenbogerd and Zobel (1999) that was listed in Section 2.6.2.

- Indexing: Full-music indexing possible as shown in the study by Downie (1999) which enables the use of the thematic catalogue model. With this, access to musical sequences found anywhere within a melody is possible. All distinct musical words in the music document collection are used as index terms. This is especially important with no knowledge of any semantic content to these 'words'.

- Index terms: It has been proven that musical words generated using the pitch dimension are useful representation of melodic information (Downie 1999). There is enough information contained within an interval-only representation of monophonic melodies that effective retrieval of music information has been achieved.

- Index data structure: Inverted files have been emphasised as being currently the best choice in data structure for indexes and this has been shown to be feasible with musical words as well (Baeza-Yates and Ribeiro-Neto 1999).

- IR Models: Numerous text search engines based on various IR models enabling ranked retrieval have been proven to be successful for large text collections. The feasibility of adapting and enhancing this technology with musical n-grams has been shown (Downie 1999; Pickens 2000).

- Fault-tolerance: Robustness of n-grams with music retrieval had been extensively investigated based on interval classification schemes (Downie 1999). With high probabilities of errors in music queries such as query-by-examples that are performances or hummings, fault-tolerance is an important aspect that is addressed in using n-grams with musical sequences. In general, the shorter n-grams will match the input query in the presence of random errors, while the longer ones will favour the ones with more accurate melodies.

- Pattern matching algorithms: Sequential matching (when query patterns have to be matched by scanning the whole collection) would not be feasible for large collections (as the matching time would grow linearly to the size of the collection) of music documents and comparative studies clearly showed that indexing would be needed for efficient retrieval (Lemström et al 1998). N-gram matching with sequential search has been shown to have a poor performance when weights were not assigned based on term relevance

(Uitdenbogerd and Zobel 1999). More sophisticated IR models would be needed. String-matching algorithms that have been proposed are for monophonic sequences (Mongeau and Sankoff 1990; Crochemore et al 2001) and not polyphonic sequences.

- Pattern induction: Most pattern extraction methods are not comprehensive and not all significant patterns are extracted. Heuristics are used and some of these are very genre specific and generally confined to tonal music. With polyphonic music, voiced information is assumed and patterns are extracted from each of the monophonic sequences. Representing a polyphonic piece in this way means that a repetition discovery algorithm will fail to identify a repeated pattern containing notes from more than one voice (Meredith et al 2002; Tseng 1999). Loss of information is reduced with the use of full-music indexing that is possible with the use n-grams.

- Music dimension representation: Pitch intervals have been useful representation of musical sequences from which n-grams for indexing can be generated (Downie 1999). No knowledge of harmony and key is needed or assumed. With advancements in transcription and OMR technologies, indexing using the pitch and rhythm would not pose a problem with the current format constraints.

2.7.2 Research Implications

In investigating the use of n-grams with polyphonic music data, some of the questions come to mind include: How can we extend the use of n-grams for polyphonic music data? Would the number of index terms from polyphonic data overwhelm the retrieval? Would rhythm information be important? How can the rhythm information be incorporated? How to best quantise performance timing deviations? *Both rhythm and polyphony are so fundamental to the nature of music that any system which ignores these facets of music information is far from ideal* (Downie 1999). The need to address rhythm has also been emphasised in Uitdenbogerd and Zobel (1999) and Clausen et al (2000).

Following are the research implications of adopting the n-gram approach to polyphonic music retrieval:

- Polyphony: The use of n-grams with music retrieval has only been based on monophonic sequences. Although polyphonic collections have been used, n-grams were generated

from monophonic sequences representing a polyphonic document. A new approach to generating n-grams from polyphonic data would be needed.

- Information content: Only the pitch dimension has been used with musical n-gram construction (Downie 1999; Uitdenbogerd and Zobel 1999; Tseng 1999). Investigation in extending the information content to include other dimensions would be useful based on the multidimensionality of musical data. The inclusion of rhythm dimension is almost a necessity with polyphonic data and this would have to be investigated.

- A priori knowledge: Key and time signature knowledge can easily be assumed from score data but this is not the case with audio. Most studies that have incorporated the rhythm dimension have assumed time signature knowledge (Clausen et al 2000; Chen and Chen 1998). Most polyphonic work has been based on the idea that music is voiced and again this knowledge cannot be assumed with all formats of data (Uitdenbogerd and Zobel 1999). Formulating an approach that enables n-grams to be generated as transparent as possible to the underlying format has to be investigated.

- Fault-tolerance: Although fault-tolerance is not critical with MIR systems, i.e., life-threatening if the system fails to perform, it is important due to the high probability of erroneous query inputs. The study by Downie (1999) concluded that the interval classification schemes investigated in addressing fault-tolerance did not seem effective. The encoding scheme where a code was assigned for every unique interval may not be feasible with polyphonic data. Interval ranges for monophonic sequences that were no larger than +24 and smaller than -24 would not be applicable for all possible intervals from an orchestral score. A new encoding function that addresses fault-tolerance and is able to discriminate relevant documents would have to be investigated.

- Value of n: Various lengths for musical n-grams have been recommended. With Downie (1999), if one is unconcerned about the possible presence of query errors, n=6 was recommended and if one wanted to maximise the fault-tolerance of a MIR system, n=4 was recommended, similar to the study by Uitdenbogerd and Zobel (1999). Tseng (1999) concluded that values greater than 3 would be needed. This would have to be investigated further with polyphonic data.

- Vocabulary size: With the 'semantic content' problem with musical n-grams, it would be difficult selecting representative terms for a music document compared to textual documents. With text, one simple possibility is to take each of the words that appears in the document and declare it verbatim to be a term and this can easily be adopted with musical n-grams. However, this tends to both enlarge the vocabulary of the collection — the number of distinct terms that appear — and increase the number of document identifiers that must be stored in the index. As with text, there are ways that each word can be transformed before being included in the index, such as case folding (conversion of upper-case letters to lower case), stemming and stop lists restrictions. This may not be necessary with musical n-grams and the vocabulary size would need to be analysed with polyphonic music data.

- IR Models: A number of IR models available with the text search engines used were adopted. Whether one model be more appropriate with the large number of possible index terms with polyphonic music data would have to be investigated.

- Term Adjacency: Although it has been studied, that the use of some type of ranked retrieval method would overcome any loss of retrieval effectiveness associated with the absence of within-song location information, thus making it unnecessary to include such information within the indexes (Downie 1999), it has been shown that structured retrieval addressing term adjacency improved the performance (Pickens 2000). This would have to be investigated further and a new approach to incorporate position information with 'polyphonic musical words', which are not only adjacent but concurrent, would need to be introduced.

- Intercepting onsets: N-grams were found not too suitable for the consolidation and fragmentation (Tseng 1999) edit operations. Problems of additional notes generating large number of n-grams differing between the query and relevant document would also be a problem with polyphonic data. N-grams generated from monophonic melodic queries would differ widely from the relevant polyphonic document if there are large numbers of intercepting accompaniment onsets. N-gramming strategies would have to be investigated to address this problem.

2.7.3 MIR Evaluation

The review in Section 2.6 has shown that a number of MIR researchers have developed test collections due to the lack of standardised evaluation and methodology for MIR research. For the methodology of this study, we similarly developed a small-scale test collection. This was either due to the collections used by the other researchers were in the monophonic form or it couldn't be obtained due to copyright problems. Details of the test collection development for this study are presented in Chapter 4.

Chapter 3

Indexing

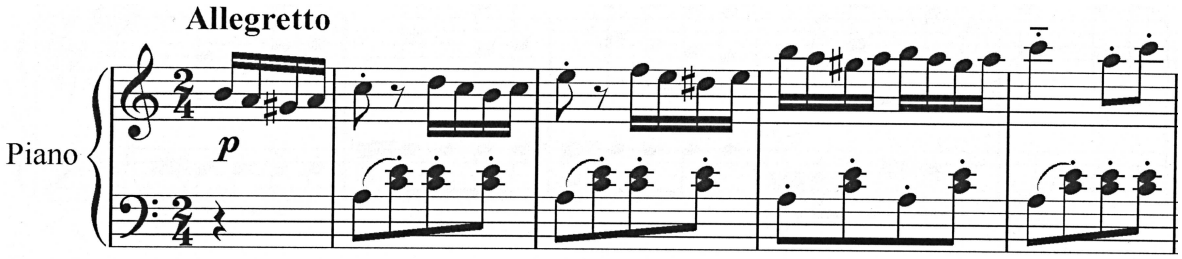
This chapter introduces an approach to constructing n-grams from polyphonic music using the pitch and rhythm dimensions of musical data. The pattern extraction and encoding approaches are described in the first two sections. These include an analysis of pitch and duration ratio distributions and the formulation of a function for the assignment of alphabetic codes to interval classes. This is followed by a discussion on the robustness with the use of n-grams for music retrieval. The last three sections outline methods to overcome problems identified with music retrieval. These include problems in general with music retrieval, and also a number of issues that are more specific to polyphonic data. Proposed solutions include the use of alternate n-grams, path selection and position indexing of adjacent and concurrent polyphonic n-grams.

3.1 Pattern Extraction

The construction of n-grams using interval-only representation of monophonic 'melodic strings' has been outlined in Chapter 1. With a gliding window, this sequence would be fragmented into overlapping length-n subsections. The n-grams generated would then be encoded as musical words that could be used as terms in a document to be indexed.

With polyphonic music data, this approach towards generating n-grams would not be applicable, since more than one note may be sounded at one point in time. This section describes a new approach towards obtaining n-grams from polyphonic music data. First, polyphonic music pieces are encoded as an ordered pair of onset time (in milliseconds) and

pitch (in MIDI semitone numbers) and these are sorted based on the onset times. There may possibly be a few different pitches corresponding to one particular onset time with polyphonic music data. We group pitches with the same or similar onset time together as musical *events*. Using the gliding window approach as illustrated in Figure 3.1, this sequence of events is divided into overlapping subsequences of n different adjacent events, each characterised by a unique onset time. The onset times used in the Figure 3.1 were extracted from a MIDI performance file of the excerpt. At time 0, 150, 300 and 450 ms, there is only one pitch event per onset time, and for onset times 600 and 900 as shown, there are two pitch events that occur for each onset. For each window, all possible monophonic pitch sequences are extracted to construct the corresponding musical words; this is discussed further in Subsection 3.1.1. Note that the term musical word in this context does not necessarily subsume a melodic line in the musical sense; it is simply a monophonic sequence extracted from a sequence of polyphonic music data.



0	71	window 1
150	69	window 2
300	68	window 3
450	69	
600	57	
	72	window 4
900	64	
	60	

Figure 3.1: Excerpt from Mozart's *Alla Turca* and the first few events with onset times and pitches

Here is the summary of this approach towards the construction of n-grams from polyphonic

music data:

- Divide the piece using a gliding window approach into overlapping windows of n different adjacent onset times;
- Obtain all possible paths of monophonic melodic strings from each window, where all paths are considered for the generation of musical words.

Various approaches in deriving patterns from unstructured polyphonic music for computer-based music analysis have been studied and categorised by Crawford et al (1998). According to this study, the approach of obtaining all possible combinations of monophonic sequences within windows of the polyphonic sequences would be termed *musically unstructured but exhaustive*. Crawford et al (1998) investigated approaches to identify patterns in musical sequences. When formulating this approach for n-gram construction, to the author's knowledge there were no studies that used all monophonic sequences from polyphonic data for indexing purposes.

Each n-gram in a musical sequence is unlikely to be a musical pattern or motive on its own, but a pattern amenable to digital string-matching. The n-grams encoded as musical words with text representations would be used in indexing, searching and retrieving a set of sequences from a polyphonic music data collection. The information content of musical n-grams have been based on the pitch dimension of music. The next two sub-sections describes how this is extended to include the rhythm dimension.

3.1.1 Pitch

In constructing musical words, an interval representation of pitches, i.e., the difference of adjacent pitch values rather than the pitch values themselves are used, similar to Downie (1999). Owing to their transposition-invariance, intervals are a common mechanism for deriving patterns from melodic strings (Lemström et al 1998). For a sequence of n pitches, a sequence of $n-1$ intervals is given by

$$\text{Interval}_i = \text{Pitch}_{i+1} - \text{Pitch}_i. \tag{3.1}$$

Figure 3.1 illustrates the pattern extraction mechanism for polyphonic music: The performance data of the first few notes of a performance of Mozart's *Alla Turca* was extracted

from a MIDI file and converted into a text format, as shown at the bottom of Figure 3.1. The left column contains the onset times sorted in ascending order, and the corresponding notes (MIDI semitone numbers) are in the right column. When using a window size of 3 onset times, one interval sequence for the first window [−2 −1] is obtained, one for the second window [−1 1] and two for the third window, [1 −12] and [1 3]. The polyphonic data in the fourth window gives rise to 4 monophonic pitch sequences within this window.

3.1.2 Rhythm

When using the rhythm dimension for music retrieval, a common mechanism is to use the relative duration of a note with respect to a designated base duration, such as the quarter or the sixteenth note. Relative durations are widely used as they are invariant to changes of tempo. However, the choice of base duration could pose quantisation problems with performance data compared to data obtained from score encodings. In order to avoid this problem we suggest using the duration ratio of consecutive notes; these are invariant to changes in tempo and robust with respect to small measurement errors.

Although MIDI files encode the duration of notes, the actual or perceived duration of notes into consideration are *not* taken into consideration, as this is nearly impossible to determine from actual performances or raw audio sources. By contrast, onset times can be identified more readily with signal processing techniques. The time between consecutive note onsets instead of note duration, which has previously been proposed and studied by Shmulevich et al (1999) are used. Independently, the use of rhythmic ratios has been investigated by Raphael (2001) using the term IOI (Inter-Onset Intervals) ratios. Summarising, we encode rhythmic information through ratios of time difference as in

$$\text{Ratio}_i = \frac{\text{Onset}_{i+2} - \text{Onset}_{i+1}}{\text{Onset}_{i+1} - \text{Onset}_i}. \tag{3.2}$$

With this approach, it is not necessary to quantise on a predetermined base duration nor to rely on the duration of a note which can be difficult to determine from audio performances. No knowledge of beat and measure information are assumed.

For a sequence of n onset times we obtain $n-2$ ratios using Eqn 3.2 and $n-1$ interval values using Eqn 3.1. An n-gram representation which incorporates both pitch and rhythm information using intervals (I) and ratios (R) would be constructed in the form of

$$[I_1\ R_1\ \ldots\ I_{n-2}\ R_{n-2}\ I_{n-1}]. \tag{3.3}$$

It has to be emphasized that the value of n here is the number of different adjacent events and not the size of the sub-string from a sequence of intervals, as discussed for the construction of n-grams from monophonic musical sequences. For a given value of n, the length of the musical word generated with this approach would be different from the word generated from the earlier method of constructing n-grams from monophonic sequences by Downie (1999). Using the example of Figure 3.1, the combined interval and ratio sequences from the first 3 windows of length 3 are $[-2\ 1\ -1]$, $[-1\ 1\ 1]$, $[1\ 1\ -12]$ and $[1\ 1\ 3]$. Note that the first and last number of each tuple are intervals while the middle number is a ratio.

3.2 Pattern Encoding

In order to be able to use text search engines n-gram patterns need to be encoded with text characters. One challenge that arises is to find an encoding mechanism that reflects the patterns we find in musical data. With large numbers of possible interval values and ratios to be encoded, and a limited number of possible text representations, classes of intervals and ratios that clearly represent a particular range of intervals and ratios without ambiguity had to be identified. For this, the frequency distribution for the directions and distances of pitch intervals and ratios of onset time differences that occur within the data set were obtained. A data collection of 3096 MIDI files of a classical music collection (http://www.classicalarchives.com) was used to compute these frequencies. A largely classical collection was used for the data analysis of this study. However, a similar analysis can be performed using other collections.

Figure 3.2 shows the frequency distribution of all occurring intervals (in units of semitones) of these 3096 MIDI files. According to this figure, the vast bulk of pitch changes occurs within one octave (i.e., with semitone differences between -12 and $+12$). Correspondingly, a good encoding should be more sensitive in this area than outside it. Using one code or text character to represent each interval in the range $\{-12, -11, \ldots, 12\}$ would clearly avoid any ambiguity. The code was chosen to be the integral part of a differentiable continuously changing function, the derivative of which closely matches the empirical distribution of intervals in Figure 3.2.

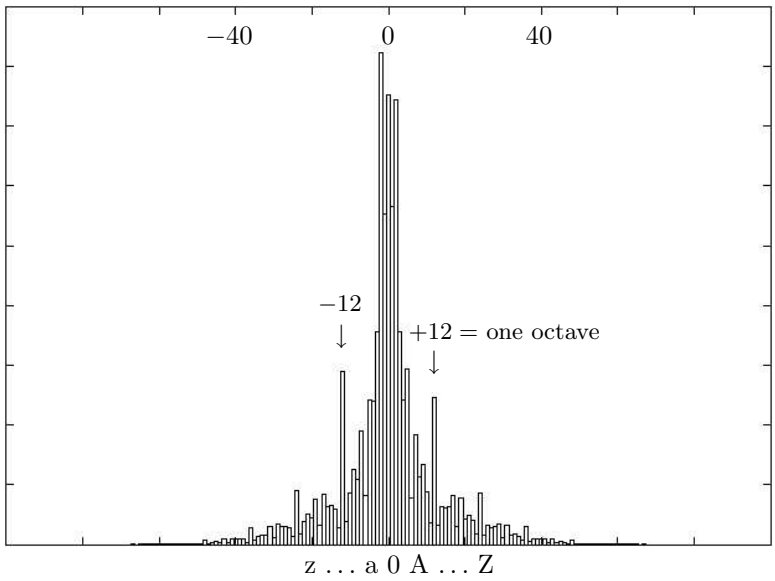


Figure 3.2: Interval histogram for 3096 classical music pieces

A suitable functional form has been found as

$$C(I) = \text{int}\left(X \tanh(I/Y)\right), \tag{3.4}$$

where X and Y are constants and $C(I)$ is the code assigned to the interval I. The function int returns the integer portion of its argument. X has the effect of limiting the number of codes, and with 26 available letters (text alphabets a-z adopted for this study), it is accordingly set to 27 in our experiments. Y is set to 24 for this achieves a 1:1 mapping of semitone differences in the range $\{-13, -12, \ldots, 13\}$. In accordance with the empirical frequency distribution of Figure 3.2, less frequent semitone differences (which are bigger in size) are squashed and have to share codes. Based on the properties of the tanh curve, Y determines the rate at which class sizes increase as interval sizes increase. This is a trade-off between classes of small (and frequent) versus large (and rare) intervals. The codes obtained are then mapped to the ASCII character values for letters. In encoding the interval direction, positive intervals are encoded as uppercase letters A−Z and negative differences are encoded with lowercase letters a−z, the code for no difference being the numeric character 0.

In using duration ratios, most studies have assumed *quantised rhythms*, i.e., as notated in the score (Shmulevich et al 1999) owing to simplicity and timing deviations that could occur with performance data. Ratio bins are adopted and are constructed as below:

Figure 3.3 shows the frequency of the logarithm of all occurring ratios of the data collection

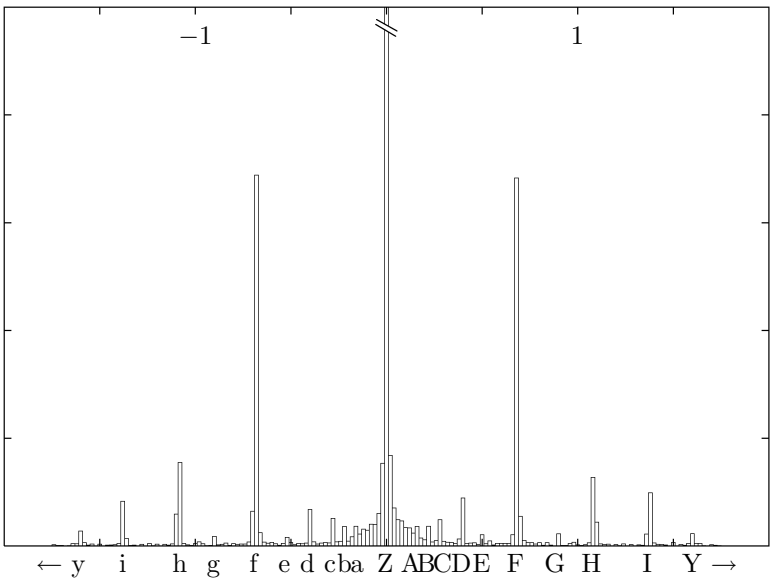


Figure 3.3: Log-ratio histogram and ratio bin labels

in the sense of Eqn 3.2. The peaks clearly discriminate ratios that are frequent. Mid-points between these peak ratios were then used as the bin boundaries which provide appropriate quantisation ranges. Ratio 1 has the highest peak, as expected, and other peaks occur in a symmetrical fashion where, for every peak ratio r, there is a symmetrical peak value of $1/r$. The proportion of ratio 1 with respect to all other entries is 34.70%. From the data analysis, the peaks identified as ratios greater than 1 are 6/5, 5/4, 4/3, 3/2, 5/3, 2, 5/2, 3, 4 and 5.

The ratio 1 is encoded as Z. The bins for ratios above 1, as listed above, are encoded with uppercase letters A−I and any ratio above 4.5 is encoded as Y. The corresponding bins for ratios smaller than 1 as listed above are encoded with lowercase letters a−i and y, respectively.

Musical words obtained from encoding the n-grams generated from polyphonic music pieces with text letters are used in the construction of index files. Queries, either monophonic or polyphonic are processed similarly. The query n-grams are used as search words in a text search engine.

3.3 Path Selection

When the window size n is large or when too many notes could be sounded simultaneously, such as in the music score shown in Figure 3.4, the number of all monophonic combinations within a window becomes large.

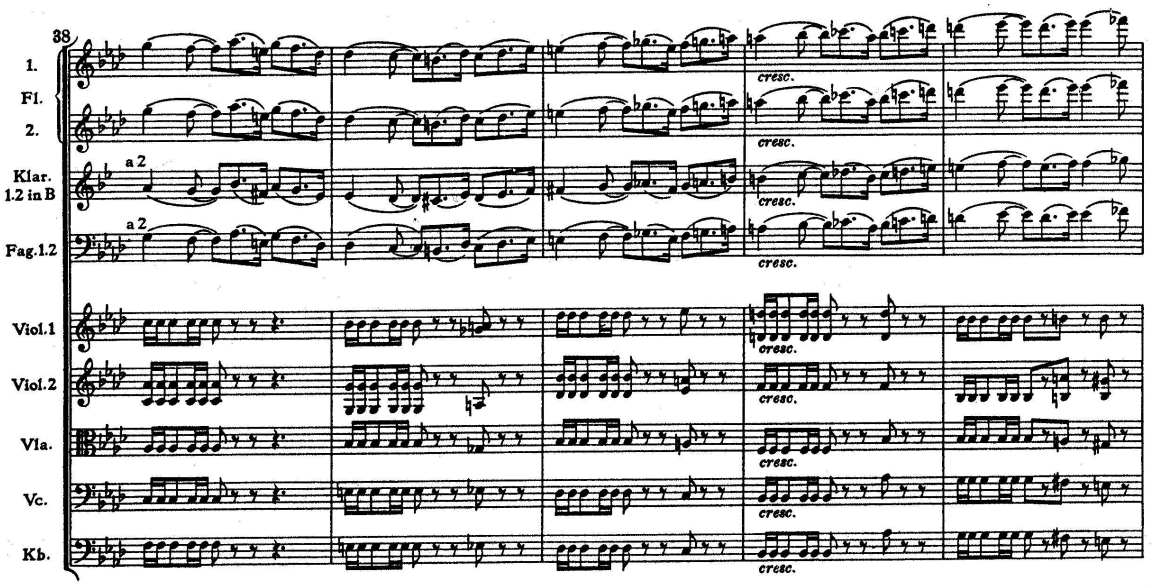


Figure 3.4: Score with large number of possible monophonic combinations — from Tchaikovsky's Fourth Symphony score in the Dover study edition, Dover Publications, Inc.

Consider, for example, the case of $n = 5$, and ten different notes played at each of the 5 onset times. As a result there would be $10^5 = 100,000$ different monophonic paths through this window: this appears to be an impractical way of indexing a tiny bit of music! This is illustrated using an example of 10 musical instruments that may be sounded together simultaneously in Figure 3.5. The black dots indicate the instrument that is being sounded for a particular onset. The example in this figure shows events for 10 onsets.

Restricting the possible combinations to variations of upper and lower *envelopes* of the window is one suggestion to restrict the number of paths. That is only to allow monophonic sequences which run through the highest two notes per event (variation of the upper envelope) or which run through the lowest two notes per event (variation of the lower envelope) as shown in Figure 3.6. For early investigation, the top two and bottom two variations are arbitrarily selected (part of melodic and part of accompaniment information). In the above example there would only be $2 \cdot 2^5 = 64$ different monophonic paths through this highly polyphonic passage of music. Although technically the restriction to envelopes still generates a number of combinations which is exponential in n, this is not a problem in practice as the sensible values for n are restricted by a tradeoff between precision and robustness. Large window sizes

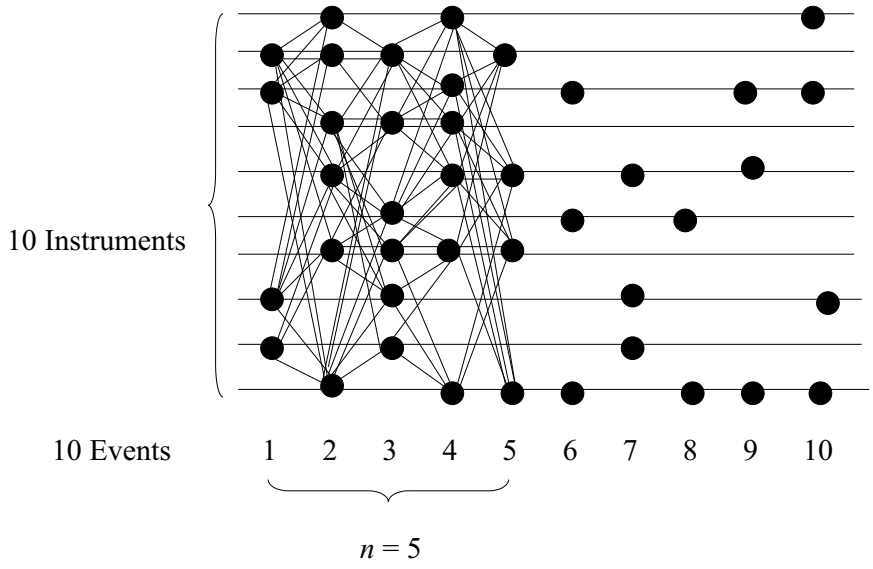


Figure 3.5: All possible paths

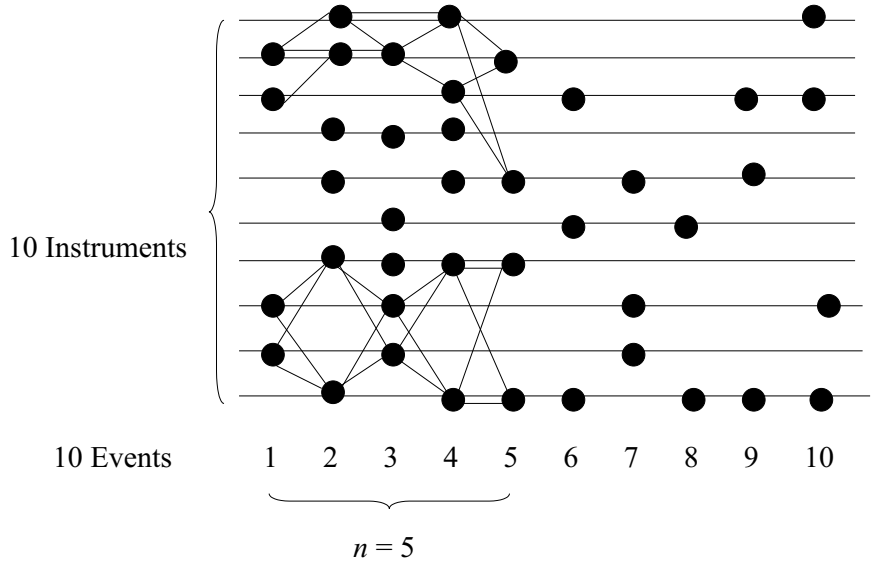


Figure 3.6: Path selection

promise better identification of the music piece but require that there be no errors such as transcription or humming errors. The shorter the window size, the more robust the queries become against errors, but the less indicative the musical words become for the music piece. Previous studies have concluded that sensible values of n are between 4 and 6 for interval-only monophonic music encoding (Downie and Nelson 2000); hence, it can expected that for interval-and-rhythm encoding of polyphonic music n should be around 3 or 4.

3.4 N-grams and Fault-Tolerance

With queries generated from erroneous inputs, such as in QBH, correct retrieval is possible using the n-gram approach to music retrieval. An erroneous query string would generate a number of n-grams that are incorrect out of the total number of n-grams constructed. The probability of retrieving a query correctly would depend on the number of query n-grams that are incorrect. Using McNab's error model described below and the value of $n = 3$ (value of n in the context of a monophonic sequence), the fault-tolerance of the n-gram approach is illustrated using the theme from Mozart's Variations in C, K265, *Ah! Vous dirai-je, Maman* (Twinkle, Twinkle, Little Star), adapted from Barlow and Morgenstern (1949) as shown in the top part of Figure 3.7. A simple monophonic sequence is selected to illustrate the fault-tolerance.

McNab's error model is based on a study by McNab et al (1996). In this experimental study towards the development of a QBH system, ten songs and ten singers were used to get an idea of the kind of input one could expect in a music retrieval system — to find out how people sang well-known tunes. The types of errors had been heuristically classified by Downie (1999) into four classes: (i) *expansion* — a tendency to expand smaller intervals that fell within 1 and 4 semitones; (ii) *compression* — a tendency to compress larger intervals that were larger than 5 semitones; (iii) *repetition* — a tendency to incorrectly repeat notes; (iv) *omission* — a tendency to simply omit a note. Using the excerpt in Figure 3.7 (a), these classes of errors are illustrated in sections (b) and (c) of the figure.

An interval-only representation for n-grams of the excerpt in Figure 3.7 (a) is deployed; they are constructed based on the interval distance (in semitones) and direction using Eqn 3.1. Therefore, the first n-gram of the top part of Figure 3.7 is [0 +7 0]. With the gliding window approach, n-grams would be repeatedly generated in this pattern to the end of the excerpt.

The set of 3-grams generated from the top of Figure 3.7 are: [0 +7 0], [+7 0 +2], [0 +2 −2], [+2 0 −2], [0 −2 −2], [−2 0 −2], [0 −2 0], [−2 0 −1], [0 −1 0], [−1 0 −2], [0 −2 0], [−2 0 +2] and [0 +2 −4].

To illustrate the fault-tolerance, two examples of possible errors, compression and omission, are incorporated into the query string as shown in Figure 3.7 (b).

The set of 3-grams generated from this are: [0 +6 0], [+6 0 +3], [0 +3 0], [+3 0 −2], [0 −2 0], [−2 0 −2], [0 −2 0], [−2 0 −1], [0 −1 0], [−1 0 −2], [0 −2 0] and [−2 0 −2].

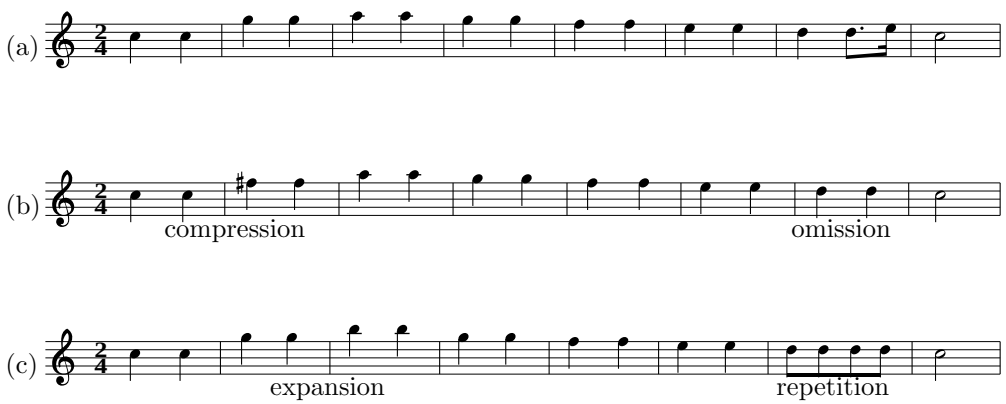


Figure 3.7: (a) Theme from "Ah! Vous dirai-je, Maman"; (b) and (c) with humming errors

In this example, around 50% of the n-grams generated from the erroneous input of Figure 3.7 (c) coincide with those from the excerpt in part (a). If this number of n-grams still sufficiently represents the indexed relevant document unambiguously, perfect retrieval is highly feasible.

With the possible errors from the example above, in using a 2:1 mapping (by assigning 48 to Y from Eqn 3.4), the intervals +6 and +7 would have been encoded with the same text letter. This would have been more fault-tolerant towards the compression error in Figure 3.7. This comes at a cost of being otherwise less discriminating.

More detailed interval classification schemes were investigated by Downie (1999), and he concluded that the expected fault-tolerance through the application of these schemes was not evident. This is in contrast to the QBH study performed by Prechelt and Typke (2001), where Parson's code (Parsons 1975) (codes that denote only the direction of intervals, either Up (U), Down (D) and Same (S)) was said to be highly fault-tolerant on a database of monophonic themes. However, in our preliminary tests retrieving polyphonic pieces using the n-grams approach with Parson's code, hardly any queries were correctly answered. Hence, this approach was not adopted either.

Rhythmic ratios can be compressed in a similar way by defining wider ratio bins (merging A and B, C and D, etc.); this is going to be used to investigate fault-tolerance with rhythmic deviations.

3.5 Alternate Onsets

Figure 3.8: Monophonic query example

With full-music indexing of polyphonic documents, n-grams generated would include n-grams from accompaniment notes. These n-grams would be matched against n-grams generated from a simple melody line such as the melodic query excerpt extracted from Figure 3.1, shown in Figure 3.8. The accompaniment might intersperse additional onset times, and hence, a match could fail. One potential strategy to overcome the problem of intercepting accompaniment onsets is to try indexing only every other onset. The number of onsets to alternate may depend on the type of accompaniment, however as a preliminary investigation of this strategy, one and two onsets are skipped.

3.6 Polyphonic Position Indexing

Proximity-based retrieval has been widely used with text and adopting its use with music data has been very limited — a preliminary study was performed by Pickens (2000) using monophonic musical sequences. Polyphonic music would require a new approach towards indexing position information using 'overlaying' positions of polyphonic musical words. This would take into consideration the time-dependent aspect of polyphonic musical words compared to indexing a 'bag of terms'. In using the n-gram approach towards indexing polyphonic music, apart from the adjacent musical words generated based on a time line, words may be generated concurrently at a particular point in time. This emphasizes the need for a 'polyphonic musical word position indexer'. Using all possible combinations of monophonic sequences from polyphonic music data, 'overlaying' word locations (when more than one word can assume the same word position) have to be included in the index so that the concurrency and sequencing information is not lost. This section first discussed the inclusion of within-document position information with polyphonic musical words. Next, the retrieval process is discussed. A new proximity-based operator for retrieval is proposed, and the similarity computation or ranking

function for this is formulated.

Using the first five onsets of the music excerpt in Figure 3.1, the document generated from our polyphonic text document generator is shown in Figure 3.9. The polyphonic text document generator is the tool developed for the conversion of a polyphonic MIDI file into a polyphonic musical text document. Modules for n-gram construction and pattern encoding with our approaches presented in Sections 3.1 and 3.2 were added to an existing midi-to-text utility developed by GN MIDI Solutions (Nagler 1998), for which source codes in C++ were obtained with kind permission. In incorporating term position information with the index, the existing parser of Lemur Toolkit (2001) infers the position of a particular term relative to the first term of the document. Lemur Toolkit (2001), discussed in Chapter 2 is the research based text retrieval toolkit adopted for our MIR study.

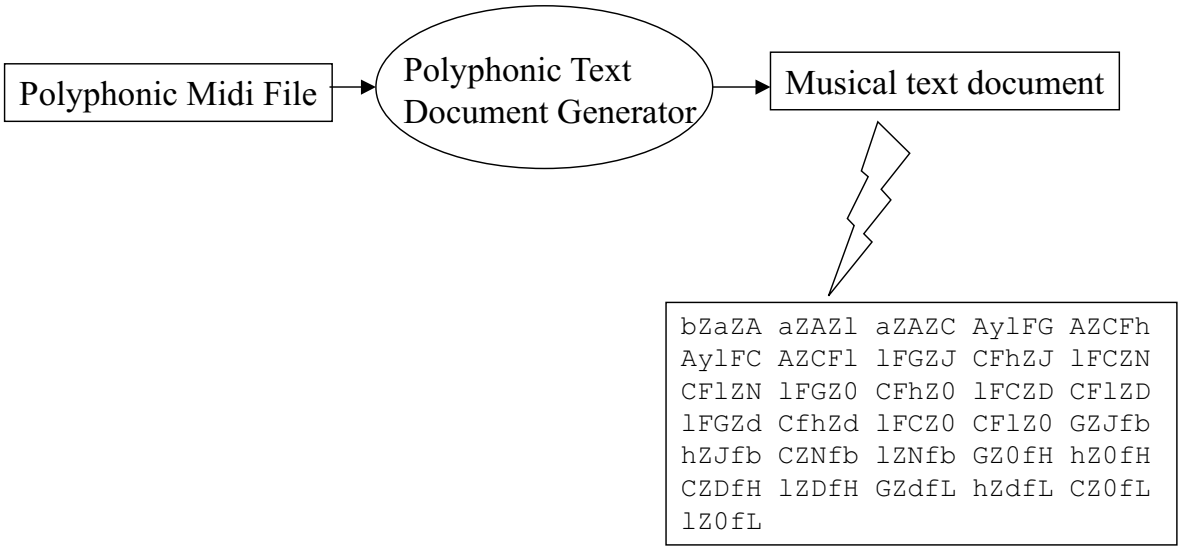


Figure 3.9: Musical text document

Existing text search engines that include position information upon indexing only consider the adjacency of text words. These are not suitable to be adapted for polyphonic music retrieval since not only adjacency has to be considered but concurrency as well. However, with large numbers of modules within this search engine still useful for indexing musical words, enhancements to the existing system for a preliminary investigation of 'overlaying' musical words was considered feasible for this study. In developing a 'polyphonic word position indexer', enhancements to the indexing module included the reading of position information from the musical text document as opposed to automatic increment of within-document po-

sition for every term in the document Lemur Toolkit (2001). The polyphonic text document generator had to be enhanced as well to indicate concurrent position information (if any) when generating musical words as shown in Figure 3.10. The parser would parse these as position information and during the indexing process, the index would then record this 'overlaying' location information.

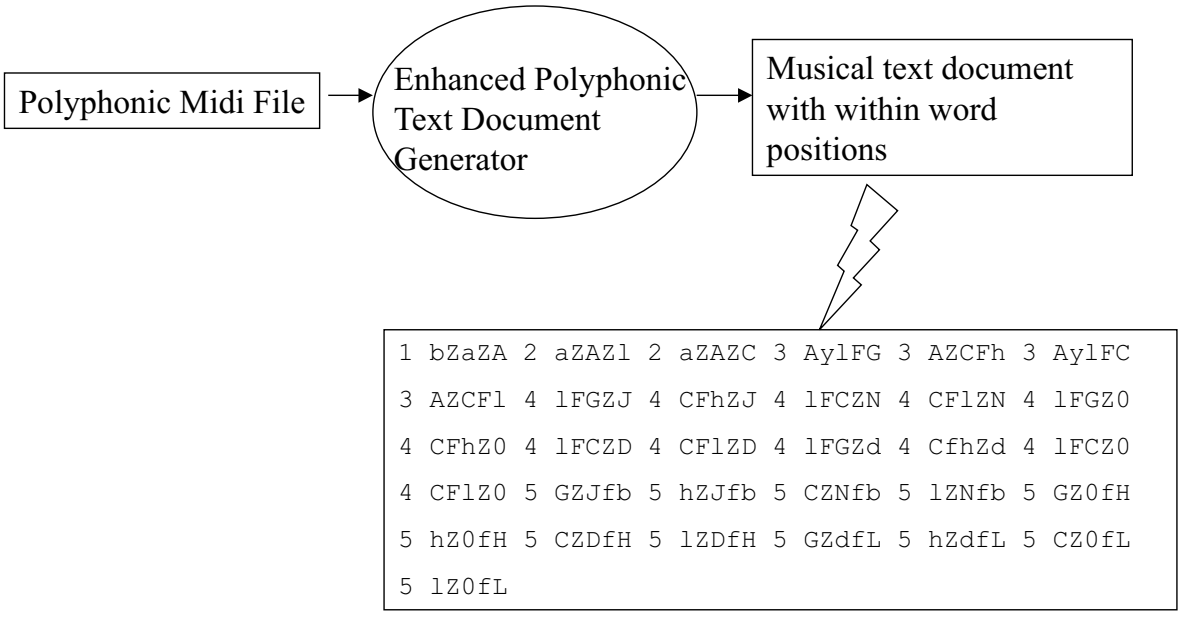


Figure 3.10: Musical text document with within-document word positions

For proximity-based retrieval, the structured query and proximity-based operators available with Lemur were first investigated for musical words. The Lemur structured query language enables the use of proximity operators (ordered and unordered windows). As discussed in Section 2.3.2, retrieval models which combine information on text content with information on the document structure are called structured text retrieval models. Query languages allow a user to combine the specification of strings (or patterns) with the specification of structural components of the document. Richer expressions for query formulations such as defining proximity ranges for retrieval phrase queries are possible. Following is the list of operators available within Lemur (the symbol # is part of the syntax for the use of these operators in the query document as required by Lemur Toolkit (2001):

Sum Operator: #sum $(T_1 \ldots T_n)$ The terms or nodes contained in the sum operator are treated as having equal influence on the final result. The belief values provided by the arguments of the sum are averaged to produce the belief value of the #sum node.

Ordered Distance Operator: #odN ($T_1 \ldots T_n$) The terms within an ODN operator must be found in any order within a window of N words of each other in the text in order to contribute to the document's belief value.

Un-ordered Window Operator: #uwN ($T_1 \ldots T_n$) The terms contained in a UWN operator must be found in any order within a window of N words in order for this operator to contribute to the belief value of the document.

Phrase Operator: #phrase ($T_1 \ldots T_n$) The operator is treated as an ordered distance operator of 3 (#od3). Note that this is a simplification of the more complicated heuristic used by Inquery.

And Operator: #and ($T_1 \ldots T_n$) The more terms contained in the AND operator which are found in a document, the higher the belief value of the document.

Boolean And Operator: #band ($T_1 \ldots T_n$) All of the terms within a BAND operator must be found in a document in order for this operator to contribute to the belief value of that document.

Or Operator: #or ($T_1 \ldots T_n$) One of terms within the OR operator must be found in a document for that document to get credit for this operator.

With classic Boolean systems, no ranking is usually provided, and a document either satisfies the query or not. The condition would have to be relaxed to overcome this limitation whereby a document that partially satisfies an AND condition might be retrieved. From the list of operators above, AND is a 'fuzzy Boolean' (a term adapted from Baeza-Yates and Ribeiro-Neto (1999) to differentiate operators from the classic Boolean model) operator, and BAND the classic Boolean operator. With 'fuzzy Boolean' operators, the idea is that the meaning of AND and OR can be relaxed, such that instead of forcing an element to appear in all the operands (AND) or at least in one of the operands (OR), they retrieve elements appearing in some operands (Baeza-Yates and Ribeiro-Neto 1999). The documents are ranked higher when they have a larger number of elements in common with the query. We push the use of 'fuzzy boolean' operators further in imposing adjacency constraints, forming a new hybrid proximity-based operator.

The existing proximity-based operator available within Lemur Toolkit (2001) that retrieves using proximity and adjacency constraints is ODN. However, using this could fail with polyphonic data, in particular with monophonic queries, due to the problems of intercepting onsets as discussed in Section 3.5. With ODN, only documents that contain *all* query terms in the similar order within a given proximity distance are retrieved. Intercepting onsets from a polyphonic document would generate n-grams that are dissimilar to its corresponding monophonic query, resulting in non-retrieval of relevant documents. Erroneous query inputs will also generate dissimialr n-grams from the relevant document. A 'musical ordered distance operator' (MODN) would need to be introduced that should enable this difference between n-grams generated from the query and the relevant polyphonic document to be reflected by a similarity measure (Doraisamy and Rüger 2003b), i.e., retrieval that partially satisfies query conditions would be needed. Ranked retrieval should be based on *the number of query terms found within a given proximity distance and not the condition that all terms must be found within a given proximity distance*. The requirement of MODN therefore would be to retrieve documents based on a rank whereby documents that match the query with the *the highest* number of query n-grams within a given proximity distance would be retrieved with the highest rank, i.e., rank 1. This would be an operator merging the definitions of the AND and ODN operators. However, Lemur's ranking approaches for these existing operators are not that easily adaptable for 'overlaying words' as discussed below.

In proposing a similarity computation approach for MODN, the ranking approaches of ODN and AND were reviewed and tested. The retrieval model used by Lemur for structured retrieval is based on the probabilistic model. This is a re-implementation of Inquery's (Callan et al 1992) structured query language, a retrieval system developed at the Centre of Intelligent Information Systems (CIIR), University of Massachusetts, USA. The model uses Bayesian inference networks to describe how text and queries should be used to identify relevant documents. Retrieval is viewed as a probabilistic inference process which compares statistical evidence of the text representations to similar evidence from the information need (Croft et al 1993). Further reading on Bayesian Inference Networks is available in Baeza-Yates and Ribeiro-Neto (1999). Inquery, like most statistical systems, relies on a $\mathrm{tf} \cdot \mathrm{idf}$ formula for estimating the probability that a document is about a concept. Unlike many systems, Inquery starts with a default probability and then adjusts it based on evidence of relevance (Broglio

et al 1994). The current belief function used by Inquery and adopted by Lemur is given below, adapted from (Allan et al 2000).

$$w_{t,d} = 0.4 + 0.6 \times \frac{\mathrm{tf}_{t,d}}{\mathrm{tf}_t + 0.5 + 1.5\frac{\mathrm{length}(d)}{\mathrm{avg\ len}}} \times \frac{\log \frac{N+0.5}{n_t}}{\log N + 1} \tag{3.5}$$

where n_t is the number of documents containing term t, N is the number of documents in the collection, "avg len" is the average length (in words) of documents in the collection, length(d) is the length (in words) of document d, and $\mathrm{tf}_{t,d}$ is the number of times term t occurs in document d. The tf component is usually referred to as an 'Okapi tf' function (Robertson et al 1994), and the idf component is the normalised idf used by the CIIR for years (as given in Section 2.3.3) (Allan et al 2000).

Initial investigation of the scoring functions revealed rather skewed results and this can be said to be due to the problem of the document length definition with polyphony. For a document with four onsets and six terms, such as 1 a 2 b 2 c 3 d 4 e 4 f, the polyphonic document length should be four, based on the number of onsets. The numbers indicate 'overlaying positions' and the corresponding alphabets representing musical words. Adopting the use of this default belief value without further investigation into the polyphonic document length is not possible. The inference of relevance evidence may have to consider the individual monophonic sequences that make up the document as opposed to the total number of terms that the document contains. Therefore in formulating a scoring function for MODN, the process had to start from scratch looking at the notion of match points for term or phrase frequencies instead (Baeza-Yates and Ribeiro-Neto 1999).

The term match point defined by Baeza-Yates and Ribeiro-Neto (1999) refers to the position in the text of a sequence of words which matches (or satisfies) the user query. Thus, if the user specifies the simple query #ODN 10 ['President's white house'] and this string appears in three positions in the text of a document d_j within a distance of 10 terms to each other, we say that the document d_j contains three match points. With ODN, this match point value is used for the tf component of the scoring function. With MODN, we continue to look at match points but the match points for polyphonic musical documents would be the position in the text document that matches the first term available of the query sequence. Apart from assigning a match point only based on the first term of the query sequence, any of the following terms in the query sequence can assume the first position if any of the prior

terms do not exist in that particular sequence. For the score calculation, 1 point is given as a score for the first term from the query sequence that is found in a text document and 1 point for each of the following term that is found within the given proximity. For query ['President's white house'] using #ODN 10, there would not have been any documents that contained 'white house' that would have been retrieved. With #MODN 10, documents that contain all three terms within a distance of 10 would have been ranked with a score of 3, two terms within the distance of 10, score 2 and 1 term score 1.

The implementation approach first involved the use of the Boolean OR operator to initially retrieve all documents that contains any of the query terms. The final computations of the similarity measure and sorting of the ranks are done only for those records that are selected by the Boolean logic — calculating proximity distance and score assignment in our case.

<DOC q1>
MODN
3
LPAREN
1 p
2 q
3 r
RPAREN

<DOC q2>
MODN
3
LPAREN
1 p
2 q
2 r
3 s
RPAREN

Figure 3.11: Query Documents

Score calculations using 'overlaying' positions are shown using the following example. The query documents are shown in the format required by Lemur's parser in Figure 3.11. Query document q1 contains a monophonic query with terms p q r and adjacent term positions 1 2 3. Query document q2 contains a polyphonic query with terms p q r s and adjacent term positions 1 2 and 3. Terms q and r occur concurrently at position 2. An example of four relevant documents are shown in Figure 3.12 and the scores for each of these documents for each of the queries are shown next.

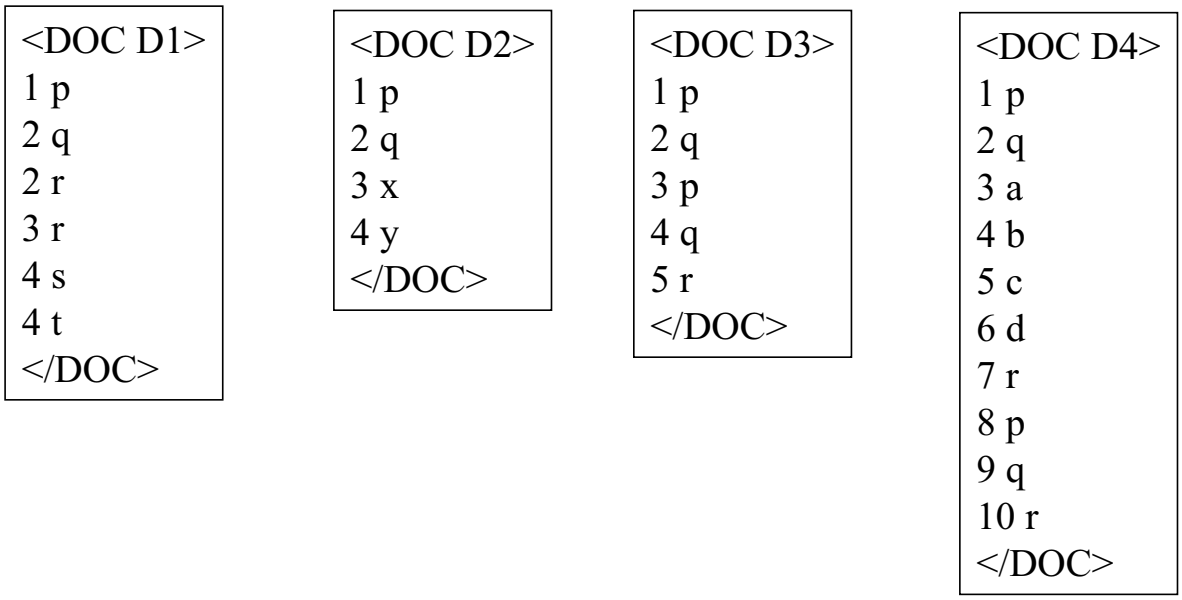


Figure 3.12: Relevant documents

The ranking scores of each of the relevant documents for q1 are:

D1 = 5 (Scores accumulated from sequences (p q r) and (p r))

D2 = 2 (Score from sequence (p q))

D3 = 6 (Scores accumulated from sequences (p q r) and (p q r))

D4 = 5 (Scores accumulated from sequences (p q) and (p q r))

The ranking scores of the relevant documents for q2 are based on two monophonic sequences – (p q s) and (p r s) and the scores are:

D1 = 9 (Scores accumulated from sequences (p q s), (p r s) and (p r s))

D2 = 2 (Score based on sequence(p q))

D3 = 4 (Scores accumulated based on sequences (p q) and (p q))

D4 = 6 (Scores accumulated based on sequences (p q), (p q) and (p r))

3.7 Summary

This chapter has outlined our approach to indexing polyphonic music using n-grams. Our strategy was to use all combinations of monophonic musical sequences from polyphonic music data. Musical words are then obtained using the n-gram approach enabling text retrieval methods to be used for polyphonic music retrieval. The n-gram technique was extended to

encode rhythmic, as well as interval information, using the ratios of onset time differences between two adjacent pairs of pitch events. In studying the precision in which intervals are to be represented, a function was formulated for mapping interval classes to text characters. With the use of bins for ranges of significant ratios, the rhythm quantisation problem in music performance data has been overcome. The robustness of n-grams with music retrieval have been discussed. Solutions have been proposed to overcome and reduce the effects of problems identified with the use of n-grams with polyphonic music. Lastly, in exploiting the time-dependent aspect of polyphonic music data, an approach enabling proximity-based retrieval was proposed and a scoring mechanism formulated.

Chapter 4

Methodology

Following the general principles of IR experimental framework design (Tague-Sutcliffe 1997; Robertson 2000), and the survey of MIR research methodologies and approaches to MIR test collections development that were presented in Chapter 2, this chapter presents the methodology for this MIR study investigating the use of n-grams for polyphonic music retrieval. Firstly, the experimental framework is presented and the discussion includes the experimental factors, the development of index files, error models, and the query acquisition and formulation approaches adopted for the search and retrieval simulation. The test collection development utilising around 10,000 polyphonic MIDI files is presented based on four stages of experimentation to evaluate various factors towards the use of n-grams for indexing and retrieving polyphonic music.

4.1 Experimental Framework

Four independent stages of experimental work were performed in this study based on the polyphonic music indexing approach and various solutions to problems that were proposed and presented in Chapter 3. Several solutions were proposed progressively after a particular stage, when problems could be identified from the evaluation and analysis of results. All variables and experimental factors used in the various stages of experimental work are summarised and presented in the following subsection.

4.1.1 Experimental Factors

Table 4.1 summarises all independent variables that were used as experimental factors for the various experimental stages.

Table 4.1: Independent variables

Factor	Definition	Codes	Comments
Dimension	The dimensions of music information contained in the musical n-gram	P	The pitch dimension
		R	The rhythm dimension
N-gram length	Number of contiguous onsets in each n-gram	3	3 contiguous onsets
		4	4 contiguous onsets
		5	5 contiguous onsets
Encoding precision	Differing levels of coarseness for encoding intervals and ratios	CP1	Y from Eqn 3.4 is set to 48 for coarser interval encoding instead of 24
		CP2	Y is set to 72 for coarser encoding than CP1
		CR	11 ratio bins are used instead of 21 for coarser ratio encoding

continued on next page

continued from previous page

Factor	Definition	Codes	Comments
Paths	Choice of possible paths from polyphonic music data for n-gram construction	AM	only one path with the highest pitch for every onset is selected
		ENV	only variations of the upper and lower envelopes are selected
Contiguity	Number of contiguous onsets that are skipped for n-gram construction	AL1	added as suffix if alternate onsets are utilised
		AL2	added as suffix if two onsets are skipped
Position	Inclusion of position information with index data	P	added as prefix with the inclusion of position information

The following questions are investigated for each factor:

- Dimension: Would the addition of the rhythm dimension to the information content of the musical n-gram improve retrieval performance? This study builds upon the study by Downie (1999) that investigated the use of n-gram for monophonic music retrieval. The lack of rhythm and polyphony has been stated as a shortcoming. The inclusion of rhythm information to the information content of the n-gram is investigated as an experimental factor in this study.

- N-gram length: Would the length of the n-gram representations affect performance? The study by Downie (1999) concludes that if one is unconcerned about the possible

presence of query errors, n=6 was recommended. If one wishes to maximize the fault-tolerance of a MIR system, the value n=4 was recommended. The value of n is based on the size of the interval string in Downie (1999) and not the number of onsets as in this study. The values $n = 4$ and 6 recommended would generate much longer musical words in the context of this study. The values of n from 3 to 5 (corresponding to values 4 to 6 in the study by Downie (1999)) are investigated in this study where n=3 would be the minimum to obtain a rhythmic ratio value.

- Encoding precision: Would coarser encodings improve fault-tolerance? The study by Downie (1999) concludes that dividing interval values into classes for encoding does improve the retrieval performance, however the fault-tolerance expected to be brought about through the application of the classification function does not appear to exist. It was recommended no to classify as there was nothing recommending the use of classification. However, this study looks again at this, but takes a different approach using a code-mapping function described in Chapter 3. The encoding precision of rhythmic information is also investigated.

- Paths: Would the selection of a number of paths compared to using all possible paths affect the retrieval performance? The study by Uitdenbogerd and Zobel (1999) is looked into for this factor. A polyphonic collection was preprocessed to obtain monophonic sequences where only the highest pitch for each onset was used for indexing. The approach proposed in this study for the selection of paths is to heuristically select paths based on the variations to the upper and lower envelopes. Retrieval performances of these strategies for path selection is evaluated.

- Contiguity: Would skipping a number of contiguous onsets overcome the problem of intersecting accompanying onsets that may occur when querying a monophonic query against a polyphonically indexed collection? Skipping one and two onsets are proposed to overcome this problem, and these are examined as experimental factors AL1 and AL2.

- Position: Would the incorporation of position information to the index information improve retrieval precision? The study by Downie (1999) concludes that the augmented $\mathrm{tf} \cdot \mathrm{idf}$ ranking methods employed affords strong retrieval performances, notwithstand-

ing the lack of within-song location information. However, an early investigation using position information and phrase operators was conducted by Pickens (2000) using monophonic data which showed positive results. Incorporation of position information is investigated as an experimental factor in this study.

On the query documents, factors investigated by Downie (1999) include the query length and the query location. Query location was not investigated in this study as it was concluded by Downie (1999) that it did not affect retrieval effectiveness. It was recommended that entire songs be indexed rather than the incipits. As for query length, it was concluded that if one is unconcerned about the presence of a query error, the shorter queries are not problematic, although longer queries are preferred. To maximise fault-tolerance, the importance of longer queries was stressed. In this study the query lengths of monophonic queries were not investigated in detail as it was also concluded in the study by McNab et al (1996) that query lengths of 12 notes or more were recommended. Owing to a lack of studies that investigate polyphonic query lengths, query lengths used in the study were adopted from Uitdenbogerd and Zobel (1999) who extracted monophonic strings from a polyphonic query. It was concluded that lengths of 30 onsets and above was recommended. It was also stressed that more than 30 onsets would also mean that melodies might be repeated within the given length.

4.1.2 Index File Development

Following the general indexing technique detailed in Chapter 3 and independent variables listed in Table 4.1, some of the parameters were varied to generate several possible index files. Queries are usually subjected to the same kind of processing. Several formats of musical words that were generated for indexing based on various combinations of independent parameters from Table 4.1 are discussed in more detail below:

P3, P4: This is a pitch-only representation of n-grams as described in Subsection 3.1.1. $n = 3$ or $n = 4$, respectively. For interval encoding, the value of Y in Eqn 3.4 is set to 24.

PR3, PR4: The pitch and rhythm dimensions are used for the n-gram construction, as described in Subsection 3.1.2. $n = 3$ or $n = 4$, respectively. For interval encoding, the value of Y in Eqn 3.4 is set to 24. For the ratio encoding, all 21 bin ranges that had been identified as significant, as listed in Subsection 3.2, were used.

CP1 as suffix: This indicates that the interval encoding is made coarser, the value of Y in Eqn 3.4 is set to 48 for a 2:1 mapping of most intervals smaller than 20 semitones (1 character now covers at least 2 semitones).

CP2 as suffix: Even coarser interval encoding, the value of Y in Eqn 3.4 is set to 72.

CR as suffix: The encoding for the ratios is made coarser: where we previously used the codes A−I, Y and a−i, y we now use the codes A−D, Y and a−d, y respectively. Now A covers what used to be represented by A and B, B covers what used to be C and D, C covers what used to be E and F, etc. There are 11 ratio bins, not 21.

ENV as suffix: The generation of n-grams is restricted to the variations of the upper and lower envelopes of the music, as discussed in Subsection 3.3.

PPR4: Incorporation of position information to PR4.

AL1 as suffix: N-grams are constructed from *every other* onset time.

AL2: Same as AL1, but n-grams were generated not from every other onset of the gliding window approach, but by skipping two onsets.

AM as suffix: Only the monophonic path through the highest notes are kept. This was motivated by a study from Uitdenbogerd and Zobel (1999) where several melodic extraction algorithms were investigated, and the approach in which the top note was extracted at each given point in time performed best. However, their study only used interval information and in this study the rhythm information is included.

To index, search and retrieve, as discussed in Section 2.3.4, the MG-1.2 text retrieval system was used in the preliminary investigation, and for all further experiments the Lemur Toolkit was used. The vector-space model is supported by both search engines, although, the main retrieval model supported by Lemur Toolkit (2001) is language model (Lafferty and Zhai 2001). Similar to the findings from the study by Pickens (2000), the language modelling approach did not perform as well in this study with musical n-grams. Initial tests were done using known-item searches. The vector-space model with the Okapi BM25 weighting performed best compared to the other tf variants – raw tf and log(tf) as well available with Lemur Toolkit (2001). Default parameter values for the BM25 weighting function provided

Index	#Possible Terms	#Terms	Proportion (%)
P3	$2,809 = 53^2$	2,803	99.79
P4	$148,877 = 53^3$	140,363	94.28
PR3	$58,989 = 53^2 \cdot 21$	57,515	97.50
PR4	$65,654,757 = 53^3 \cdot 21^2$	8,903,618	13.56
PR4AL1	$65,654,757 = 53^3 \cdot 21^2$	13,259,148	20.19
PR4AL2	$65,654,757 = 53^3 \cdot 21^2$	16,044,883	24.44
PR4AM	$65,654,757 = 53^3 \cdot 21^2$	1,530,352	2.33
PR3CP1	$58,989 = 53^2 \cdot 21$	41,028	69.55
PR3CP2	$58,989 = 53^2 \cdot 21$	26,986	45.75
PR4CP1	$65,654,757 = 53^3 \cdot 21^2$	5,697,841	8.68
PR4CP2	$65,654,757 = 53^3 \cdot 21^2$	3,293,668	5.02
PR4ENV	$65,654,757 = 53^3 \cdot 21^2$	4,350,263	6.63
PR5ENV	$73,073,744,541 = 53^4 \cdot 21^3$	17,516,717	0.02
PR3CP1CR	$30,899 = 53^2 \cdot 11$	25,480	82.46
PR3CP2CR	$30,899 = 53^2 \cdot 11$	16,818	54.43
PR4CP1CR	$18,014,117 = 53^3 \cdot 11^2$	2,684,556	14.90
PR4CP2CR	$18,014,117 = 53^3 \cdot 11^2$	1,491,649	8.28

Table 4.2: Proportion of used code space

by Lemur Toolkit (2001) were adopted for this initial study as described in Section 2.3.4. Although values adopted by Lemur is based on values recommended in Robertson et al (1994) for text retrieval, these were adopted for this first attempt in using Lemur Toolkit (2001) for MIR.

For the last stage of experimental work on proximity analysis, the structured query language provided by Lemur was investigated and we performed necessary enhancements as discussed in Section 3.6 for polyphonic music retrieval.

Table 4.2 shows a summary of the used musical word formats, listed in alphabetical order and the number of possible terms that could be generated with the variation of parameters. To obtain an estimate of the proportion of code space that was actually used, the number of unique terms was computed that were generated using 5456 files from the collection. This

proportion is shown as last column.

4.1.3 Error Models

With high probability of query imprecision or inaccuracies, predefined queries, responses and metrics for evaluation would need to be based on error models — how well a system performs under such erroneous inputs. Examples of errors are with musical inputs include that arise from singing as well as transcription. It is therefore essential for the development of robust retrieval systems that real-world query-imprecision is taken into account in the form of error models when testing the system. Error models for both monophonic and polyphonic queries were used in this experimental framework.

Monophonic Queries

For monophonic queries, QBM with the QBH interface is focused on in particular. A number of QBH studies have addressed error models and a survey of these were performed. Although the McNab error model presented in Chapter 3 was adopted in investigating the fault-tolerance of erroneous monophonic queries, other error models surveyed are discussed in this section very briefly and reasons for not adopting these are presented.

One other study that was looked at for pitch error models was the study by Haus and Pollastri (2001). The audio query transcription algorithm developed by Haus and Pollastri (2001) assumes constant sized errors based on the idea that every singer has her or his own reference tone in mind. Singers would simply sing each note relative to the scale constructed on their own reference tone, apart from some small increases with the size of the interval.

A tempo analysis was done in the QBH study by Kosugi et al (2000). It was observed that singers decided on what tempo to maintain, which was not necessarily the same as that of the original song. The assumption adopted on tempo in their study was that for faster songs there was a tendency for users to choose a tempo that was half the correct one. Fault-tolerance was addressed in the music database by making two copies of songs of fast tempos, one at the original tempo and the other at half the tempo.

Our representation of n-grams with intervals and rhythmic ratios is invariant to transpositions and scale differences, and augmentation and diminution of tempo, respectively. Hence, this representation already addresses the constant transpositions discussed by Haus and Pol-

lastri (2001) and tempo differences observed by Kosugi et al (2000).

In the studies surveyed, the error models were based on a definition of humming as singing with the syllable ta, da or la, and not whistling or singing with syllables derived from lyrics. It should be noted that Haus and Pollastri (2001) suggested the removal of lyrics to suppress another possible source of errors that is difficult to quantify. There is also the consideration that, with singing based on lyrics, one could possibly remember the tune better when it is associated with words.

Polyphonic Queries

Querying a musical database with a snippet of music played on the radio is another prominent query method. This query method, QBE also has applications for uncovering copyright infringement on the internet. For polyphonic queries, QBE is focused on in particular.

The likely errors for QBE are of a completely different kind. They result from deliberately different interpretations in different performances and from transcription errors. It is plausible that a number of independent factors contribute to deviations in pitch and timing, in which case a generic Gaussian error model seems appropriate to describe the cumulative effect of the error sources. This error model should be additive for the intervals (3.1) and additive for the logarithm of the rhythmic ratios (3.2) or, equivalently, multiplicative for the ratios:

$$\text{NewInterval}_k = \text{Interval}_k + D_i \cdot \varepsilon \tag{4.1}$$

$$\text{NewRatio}_k = \text{Ratio}_k \cdot \exp\left(D_r \cdot \varepsilon\right) \tag{4.2}$$

Here, ε is a Gaussian random variable with mean 0 and standard deviation 1, D_i is the standard deviation for an interval error and D_r is the standard deviation for an error in the ratio.

4.1.4 Query Documents

Both monophonic and polyphonic queries are used for evaluating the retrieval performance. This section discusses the various types of queries used for the search and retrieval simulation.

Known-item Searches and Simulated Errors

One of the query acquisition approaches adopted is extracting queries from documents in the collection enabling known-item searches. Two types of queries from the music pieces were extracted: *monophonic queries*, where only one note per onset time is extracted, and *polyphonic queries* where a polyphonic subsequence of events from the music piece are extracted.

Monophonic queries are thought to resemble humming, and humming errors using McNab's error model (see Subsection 4.1.3) are simulated in the following way: Each note is subjected to a certain error probability p of being altered; if a note is to be altered then compression/expansion occurs with probability 40%, repetition with probability 40% and omission with probability 20%. These values of p were adopted from the study by Downie (1999) using monophonic queries as the best approximation of McNab's error model. In the study by Downie (1999), the query length and number of notes that were simulated with errors were constant. However, with real queries possibly varying in length, we do not fix the query length. Error probability at 10% and 20% for each query note was investigated with varying query lengths.

For polyphonic queries, each interval or ratio is altered according to Eqns 4.1 and 4.2. The only relevant document for this type of query is the music piece from which the query was extracted, and not variants or otherwise similar pieces, and the quality of the retrieval mechanism is judged by the reciprocal rank of the known item.

Ad-hoc User Queries and Relevance Judgements

In order to simulate *ad-hoc queries*, where the collection is kept constant but the information need changes, we hand-crafted monophonic queries[1]. For each query, there were several performances that were considered as relevant documents. This was based on the same relevance assumption that was used by Uitdenbogerd and Zobel (1999). To make these monophonic queries realistic, we also subject each query ten times independently to McNab's error model with error probability for each note of 20%.

[1]The ad hoc retrieval task has been described to be similar to how a researcher might use a library — the collection is known but the questions likely to be asked are not known (Voorhees and Harman 1999).

Structured Query

The use of the various proximity-based and structured query operators available within Lemur Toolkit (2001) are investigated. Query formulation using all operators listed in Chapter 3 (with the exception of BAND) are tested with musical words.

A few initial tests using the known-item search with the ODN operator showed, as expected, a poor performance. Retrieval using more complex query formulations were then looked into. Such query formulations were performed by Pickens (2000) using Inquery (Callan et al 1992). According to Pickens (2000), using nested phrase operators of Inquery in a manner that attempts to recapture the original sequentiality of the song produces more precise results. With proximity-based retrieval shown to improve retrieval with monophonic data, this is further investigated with polyphony. For the nested monophonic queries, two contiguous musical words for the formation of smaller phrases within a longer query were arbitrarily selected. This query formulation towards was then used in the investigation of querying monophonic queries against a polyphonically encoded collection. A monophonic theme extracted from Figure 3.1 is encoded as:
[bZaZA aZAZC AZCIB CIBib BibZa bZaZA aZAZD AZDIA DIAia AiaZa aZaZA aZAZG AZGZb GzbZa bZaZA aZAZB AZBZb BzbZa bZaZA aZAZC]
Arbitrarily selecting two contiguous musical words for the formation of smaller phrases within a longer query, the query would be reformulated as:

```
#SUM(   #ODN3(bZaZA aZAZC)
        #ODN3(AZCIB CIBib)
        ...
        #ODN3(bZaZA aZAZC))
```

4.2 Test Collection Development

Lacking a commonly acknowledged standard test collection[2], a test collection for polyphonic music retrieval had to be designed.

[2]The candidate MIR test collection (Byrd 2000) was surveyed, however the collections were either unsuitable or difficult to obtain due to copyright problems.

A collection of almost 10,000 polyphonic MIDI performances that were mostly classical music performances had been obtained from the Internet (http://www.classicalarchives.com). These were organised by composers — Bach, Beethoven, Brahms, Byrd, Chopin, Debussy, Händel, Haydn, Liszt, Mendelssohn, Mozart, Scarlatti, Schubert, Schumann and Tchaikovsky. Other composers' works were organised in directories alphabetically ('midi-a-e', 'midi-f-m', etc.). The rest of the collection was categorised as 'aspire', 'early', 'encores' and 'others'. A smaller collection of around 1000 MIDI files of various categories of popular tunes — TV and movie themes, pop, oldies and folksongs was collected from the Internet (no longer available). Files that converted to text formats with warning messages on the validity of the MIDI file such 'no matching offset' for a particular onset, by the midi-to-text conversion utility (Nagler 1998), were not considered for the test collection. This test collection was divided into various subsets – training and test sets for the various experiments. Experimentation using training and test sets are important to avoid over-training or over-fitting when optimising parameters. The retrieval performances of retrieval runs when replicated on a different data set, should not differ too much on a new data set compared to the training set on which initial test runs were performed.

The rest of this section describes the test collection development utilising various subsets of the MIDI file collection for each of the fours stages of experimentation. As discussed in Chapter 2, a test collection is made up of a set of documents, queries and relevance judgements. These components are discussed for the test collections of every experimental stage.

4.2.1 Experiment 1: Preliminary Investigation

A preliminary study on the feasibility of the approach for the construction of n-grams from polyphonic music data towards full-music indexing was performed using 3096 files of the collection. This subset of files, around a third of our file collection had been used for the data analysis of the frequency distribution of intervals and ratios discussed in Chapter 3.

For known-item searches polyphonic excerpts were extracted from 30 randomly selected musical documents similar to the study by Downie (1999). The only relevant document for this type of query is the music piece from which the query was extracted, and not variants or otherwise similar pieces. In order to vary query lengths, they were set to 10, 30 and 50 onset times. Query lengths investigated with monophonic studies have been up to 12 notes

(McNab et al 1996; Downie 1999). According to McNab et al (1996), approximate search, in general, requires twelve notes or more to keep the number of retrieved songs manageable. Uitdenbogerd and Zobel (1999) used 10, 30 and 100 onsets for retrieval from a collection of monophonic sequences extracted from their polyphonic collection.

Using the Gaussian error model and the MRR measure for the retrieval performance measure, this test collection developed enabled examining the retrieval effectiveness of the musical words. A polyphonic MIR setting was simulated that, given a polyphonic excerpt, retrieves all pieces that contain this excerpt. Details of the experimental factors investigated are discussed in the following chapter.

4.2.2 Experiment 2: Comparative and Fault-tolerance Study

The second experiment was performed to test the feasibility of querying a polyphonic music collection with a monophonic sequence (Doraisamy and Rüger 2002). In particular, we focus on QBH systems and the fault-tolerance of the n-gram approach was examined based on QBH error models. In order to simulate ad-hoc queries, we hand-crafted ten monophonic queries. These were popular tunes of various genres. The list of songs and relevant documents are listed in Table 4.3.

With pieces from the classical collection (Songs ID 1, 4, 6, 7, 8 and 10), using the filename and composer directory, one performance of each tune was identified. Each of these performance files was edited using a midi sequencer, jazz-4.1.3, to extract a polyphonic excerpt containing the theme. Using the retrieval approach and the optimal parameters identified from the first experiment, these polyphonic excerpts containing the theme were used as queries to form a relevant document pool. MIR studies that include this small-scale pooling approach to relevant document acquisition includes the studies by Uitdenbogerd (2002) and Sødring and Smeaton (2002). The documents retrieved were listened to and its relevance was judged based on assumptions similar to Uitdenbogerd (2002). These assumptions were listed in Section 2.6.2. This provided a list of relevant documents for each polyphonic query. The theme from these polyphonic sequences were extracted manually in order to obtain monophonic queries. Midi to text utilities were used for the extraction of the monophonic sequence and text to midi conversion was performed to obtain the queries as MIDI performances.

With the popular music pieces (Songs ID 3 and 9), only one version of each was available

in the collection. These were used to extract monophonic queries using the midi sequencer. Versions for these were obtained from the Internet using the same relevance assumption. Lastly, Happy Birthday (Song ID 2), assumed to be a tune that everybody knew, was added. This was not available in our collection at all, therefore as many versions possible, based on the relevance assumption defined above, were obtained from the Internet and one of the versions with basic chords accompanying the tune was selected to extract the monophonic query. Query lengths varied between 15–25 notes for eight of the songs. The query for Beethoven's Symphony No. 5 had just 8 notes and that for Hallelujah was the most elaborate with 285 notes.

Song ID	Song Title	No. relevant
1	Alla Turca (Mozart)	5
2	Happy Birthday	4
3	Chariots of Fire	3
4	Etude No. 3 (Chopin)	1
5	Eine Kleine Nachtmusik (Mozart)	5
6	Symphony No. 5 in C Minor, (Beethoven)	8
7	WTC 1, Fugue 1, Bk 1 (Bach)	2
8	Für Elise (Beethoven)	3
9	Country Gardens	2
10	Hallelujah (Händel)	7

Table 4.3: Song list

The detailed relevant document list (query-relevant set, qrels) for these queries is available in Appendix A. It is listed in a format amenable to the performance evaluation routines that were used in the study. This relevance list can be reused for any future retrieval evaluation on this test collection.

For performance evaluation, the precision-at-15 measure was used, similar to the study by Uitdenbogerd and Zobel (1999) from which the relevance assumptions for this study was adopted. 5380 files from randomly selected subdirectories of the polyphonic music test collection formed the document set for this experiment.

4.2.3 Experiment 3: Robustness and Path Selection

This investigation is to evaluate the retrieval performance when the number of index terms generated are restricted to the upper and lower envelopes. This also includes a more extensive investigation on the robustness of the QBH and QBE tasks (Doraisamy and Rüger 2003c). Both monophonic and polyphonic queries are used for this. The approach of extracting the highest pitch from the several possible pitches for each onset was the best of the several melody extraction algorithm investigated by Uitdenbogerd (2002) and was therefore used to extract the monophonic queries as query melodies. 6366 files were used as the document set for this experiment. These files were subjected to a random subdivision between a training set and two test sets, the former to experiment with the parameters of the retrieval algorithms, the latter to be used in future for replicating retrieval runs with optimal values for parameters reported in the conclusion of this study. The retrieval performances of *known-item searches*, i.e., where queries were extracted from parts of randomly selected ten music pieces in the collection are reported in Section 5.1.3. The quality of the retrieval was judged using the MRR measure. Query lengths were fixed as 30 onsets as discussed in Section 4.1.1.

4.2.4 Experiment 4: Proximity Analysis

This section describes the test collection development of the work on proximity analysis. Term position information with indexes is known to improve the retrieval performance (Baeza-Yates and Ribeiro-Neto 1999). Query formulation using proximity operators available with the structured query language of Lemur Toolkit (2001) is investigated.

In enhancing the test collection already developed thus far, the query set was extended to include 50 queries (40 added to the 10 used in the third experiment). This list of queries and relevant documents is given in Appendix B. This number of queries were selected based on the TREC experimental model, where 50 topics are listed and queries are generated from these. TREC uses 25 topics as a minimum and 50 topics as the norm (Voorhees and Buckley 2002). These additional queries were also considered to be highly likely real-life queries for a classical collection as queries as these were amongst the 'pop' of classical music. Although the query list might appear to be biased towards popular queries, this query list can easily be extended to a more comprehensive list, possibly including real-world queries such as collection of queries from music libraries, music stores, non-Western music, etc. (Downie 2003b).

The *Dictionary of Musical Themes* (Barlow and Morgenstern 1949) that contains over 10,000 entries is certainly a useful repository that is already available for experimental query simulation. The query list, that comprises works of various composers and periods of music within the scope of tonal music, is deemed sufficiently comprehensive for the QBM task. A separate task may need to be defined for contemporary music with its own set of specific problems, such as difficulty to define a melody in this class of music (Bonardi 2000).

Exhaustive judging, where relevance is determined for each document, is feasible for a collection of this size. The relevance assumption of versions used by Uitdenbogerd (2002) was adopted and the first few seconds of each performance were listened to for a judgement to be made. This seemed a reasonable approach, as the author was sufficiently familiar with the query melodies to be able to recognise a performance within the first few seconds.

For this preliminary investigation on proximity analysis, twenty-five queries from the query document list were reformulated using the various proximity-based and structured operators. The operators used are presented in the following chapter. With these queries being an addition to the list of ad-hoc queries used in Experiment 2, these and the corresponding relevant documents were added to the same subcollection of Experiment 2 as well, bringing the number of files in this subcollection to 5456 files. For performance evaluation, the MRR measure was used. Although normally used with the known-item search, the MRR measure adopted in our previous experiments was used in this context based on the best rank of relevant documents retrieved at precision-at-15.

4.3 Summary

An experimental framework has been outlined using a test collection that we developed. Various subcollections, queries, relevance judgements and evaluation metrics were adopted for the different levels of experimental work.

Chapter 5

Evaluation Results

This chapter presents the results of the experiments performed to evaluate the new approach to indexing polyphonic music. The results are discussed independently for the four stages of experimental work done. This is followed by a summary of findings based on these results.

5.1 Experimental Stages

This section presents the results for each of the four independent stages of experimental work. The test collection setup, i.e., the set of documents, queries and relevance judgements for each corresponding stage was presented in Chapter 4 and the various evaluation metrics adopted for the evaluation has been described in Chapter 2. The implementation results are as follows:

5.1.1 Experiment 1

This initial experiment performed was to examine the retrieval effectiveness of the approach proposed for musical n-gram construction and the encoding of these for musical words generation. Two retrieval runs, Run1 and Run2, were executed investigating various experimental factors.

For the initial retrieval run, Run1, six of the musical n-gram variants listed in Table 4.2 were used in indexing the collection. These were: P4, R4, PR3, PR4, PR4CP1 and PR4CP2. The retrieval performances with the MRR measure (discussed in Section 2.6.3) are shown in Table 5.1. The MRR measure was an average of reciprocal ranks over 30 queries.

The results clearly indicate that using n-grams with polyphonic music retrieval is a promis-

	10	30	50
P4	0.60	0.77	0.81
R4	0.03	0.11	0.15
PR3	0.46	0.74	0.81
PR4	0.74	0.90	0.95
PR4CP1	0.71	0.83	0.71
PR4CP2	0.47	0.68	0.73

Table 5.1: MRR measures for Run1 with perfect queries

ing approach. The best retrieval measure 0.95 was obtained by musical words of the PR4 format and a query length of 50 onset times. Comparing the retrieval measures of P4 and PR4 for all 3 query lengths, it is clear that the addition of rhythm information to the n-gram is a definite improvement to widen the scope of n-gram usage in MIR.

The length of a window for n-gram construction requires further study, as there are clear improvements of measures between PR3 and PR4 for all query lengths. Further experiments will be needed to obtain the optimal length. In looking at the class size of intervals and the bin range of ratios, measures clearly deteriorate from smaller class sizes of PR4 to larger sizes of PR4CP1 and PR4CP2. Class sizes require further investigation to determine their usefulness in providing allowances for more fault-tolerant retrieval.

In general, and as expected, the measure improves with the length of the query for all databases, although retrieval using only ratio information with R4 is almost insignificant. With R4, there are only 441 possible number of index terms ($21 \cdot 21$ — using 21 ratio bins). These were insufficient to discriminate the pieces.

A second run, Run2, was performed by simulating errors in the queries. This was to study the retrieval behaviour under error conditions using the Gaussian error model presented in Section 4.1.3. With this initial attempt to investigate retrieval with error conditions, two sets of error deviation values D1 and D2 were arbitrarily selected as there were no particular error models on polyphonic performances or transcription that the author had knowledge of at the time of the study. With D1, D_i was assigned 3 and D_r was assigned 0.3 from Eqns 4.1 and 4.2. For the second set of mean error deviation values, D_i was assigned 2 and D_r was retained as 0.3. D_r was left unchanged, as the ratio bin range was not varied between PR4CP1 and

PR4CP2. All musical words generated for the same queries as Run1, and with length 30, were modified by incorporating the error deviation for the pitch and duration dimensions. This was done for the 3 databases PR4, PR4CP1 and PR4CP2. Only length 30 was selected for the second run as retrieval performances did not improve significantly between query lengths of 30 and 50 onsets for most formats. Also, it was stated in Uitdenbogerd and Zobel (1999) that 30 onsets were usually sufficient to represent a query melody. The MRR measures are shown in Table 5.2.

	D1	D2
PR4	0.24	0.50
PR4CP1	0.30	0.65
PR4CP2	0.27	0.50

Table 5.2: MRR measures for Run2 with erroneous queries

The results clearly indicate that musical words encoded with a wider interval class size perform better under error conditions. This can be seen from the improvement in measures obtained with Run2 and deviation set D2 of Table 5.2 where the measure of PR4CP1 is 0.65 and PR4 only 0.50. For the corresponding run, Run1, with no query errors, it shows deterioration in the measure with the wider encoding (a measure of 0.90 was obtained with PR4 and only 0.83 for PR4CP1 with query length 30). A compromise is clearly required between musical words encoded using larger interval class sizes, wider ratio bin ranges and smaller ones. This initial experiment under error conditions indicates that the use of coarser interval encoding affects the retrieval performance.

5.1.2 Experiment 2

This second level of experimentation is a comparative and fault-tolerance investigation. The investigation also looks into problems that arise from querying with a monophonic query against an indexed polyphonic collection. For the performance evaluation, the precision-at-15 measure was used, in which the performance of a system is measured by the number of relevant melodies amongst the first k retrieved, with k=15 in this case. The MIR study by Uitdenbogerd and Zobel (1999) selected the value k=20. However, 15 was selected, since quite often users may not look at documents retrieved too low in rank. Three runs were performed,

the first with perfect queries and the remaining two with erroneous queries. Detailed results for each these three runs are given in Tables C.1, C.2 and C.3 in Appendix C. The tables show the percentage retrieved up to rank 15 with respect to the number of relevant documents for each song. The last row of weighted averages (W.A.) in each of these three tables summarises the retrieval performances of the 10 queries listed in Table 4.3 as an average, weighted by the number of relevant documents for each query. The values have been rounded to the nearest integer. Table 5.3 summarises Tables C.1, C.2 and C.3 by listing the W.A. values for each of the runs.

Word format	Perfect	10%	20%
PR4	58	43	38
PR4CP1	18	14	9
PR4AL1	40	39	32
PR4AL2	34	25	10
PR4AM	80	70	58
P4	8	3	0

Table 5.3: Weighted averages at rank 15 for retrieval of queries with different error levels

The 10 queries used in the initial retrieval run were error free. The weighted average of the precision-at-15 measures for these 10 queries are shown under the Perfect column in Table 5.3. In querying a polyphonic database with a monophonic query, it is clear from the results that preprocessing a polyphonic database for indexing with a melody extraction algorithm is a feasible approach. This had been investigated previously by Uitdenbogerd and Zobel (1999) using the pitch dimension as AM. The rhythm dimension was added in this study for PR4AM. On average, 80% of the relevant documents were retrieved within rank 15 with PR4AM.

However, looking at full-music indexing of polyphonic music, PR4 performed well despite the large number of index terms generated from n-gramming all possible patterns of polyphonic music data. 58% of the relevant documents were retrieved on average within rank 15. The performance of PR4 clearly indicates that using n-grams in querying a polyphonically encoded database with a monophonic query is feasible and promising. Ways to improve the performance of PR4 could be investigated further as there is no loss of information with full-music indexing of PR4.

In comparing the individual retrieval performance of the various songs between PR4AM and PR4 in Table C.1, two of the largest retrieval measure differences were of Songs 4 and 6. Song 4 had a large number of accompaniment notes interleaved between the melody lines in comparison to the other songs retrieved perfectly by PR4AM, namely songs 1, 5 and 7. This is one possible reason for the poor retrieval of this song. This requires further investigation. The query lengths for Songs 6, of just 8 notes, were just not sufficient for retrieval of this large movement of a symphony.

The problem of intercepting accompaniment onsets was not overcome by alternating n-grams with PR4AL1 and PR4AL2. Skipping more onsets will have to be investigated in the future. Other songs not retrieved based on this problem were Song 3 and versions of songs 2, 8, 9 and 10. With PR4AL2, short query length as with Song 6 posed a problem where no query document could possibly be generated. The addition of the rhythm dimension clearly improves retrieval, as can be seen from the weak retrieval performance of P4 in comparison to PR4.

Fault-tolerance investigation was performed by simulating errors in the queries with the probability of error levels at 10% and 20% for each of the query notes. The erroneous notes were simulated using the McNab et al (1997) error model discussed in Chapter 3. The retrieval performance measures were obtained by averaging the performance results of ten retrieval runs for each song. The results are shown in Table 5.3 under the columns 10% and 20% respectively. Detailed song to song measures are given in Tables C.2 and C.3.

The retrieval performance for all databases deteriorated under the error conditions as expected with the increase of erroneous notes. It is also clear from the results that retrieval was not completely lost due to erroneous query notes with the n-gram approach, as discussed in Chapter 3. The performance of PR4AL1 remained almost similar at 40% and 39% under perfect and error conditions (error probability of 10%) respectively. More extensive tests would be required to investigate reasons for this fault-tolerance.

These results show that the simple approach of indexing only n-grams generated from the highest pitch at each onset with PR4AM retrieves best. The results also show that using all paths with PR4 for full-indexing of polyphonic music can be successfully deployed for a range of ad-hoc queries over varying genre with a reasonable expectation of finding relevant documents among the first 15 retrieved.

5.1.3 Experiment 3

This experiment continues to investigate the fault-tolerance of the n-gram method. The problem of large numbers of index terms for larger values of n is investigated by indexing selected paths of all possible polyphonic paths for n-grams generation. Both monophonic and polyphonic queries were used.

Monophonic Queries

Table 5.4 shows the mean reciprocal ranks of monophonic queries with several indexing and retrieval methods, The training set of 2122 files, discussed in Section 4.2.3., were used. The reason for the relatively large standard deviation of the averaged values lies in the nature of Information Retrieval tasks: usually, there are 'difficult' and 'easy' queries.

Index	perfect query	$p = 10\%$	$p = 20\%$	$p = 30\%$	$p = 50\%$
P3	0.05 ± 0.15	0.03 ± 0.11	0.03 ± 0.12	0.02 ± 0.10	0.03 ± 0.11
PR3	0.31 ± 0.45	0.19 ± 0.35	0.16 ± 0.31	0.15 ± 0.30	0.10 ± 0.25
PR3CP1	0.17 ± 0.31	0.19 ± 0.36	0.17 ± 0.34	0.16 ± 0.34	0.10 ± 0.26
PR3CP2	0.13 ± 0.30	0.10 ± 0.26	0.09 ± 0.24	0.10 ± 0.26	0.07 ± 0.21
PR3ENV	0.39 ± 0.43	0.27 ± 0.38	0.23 ± 0.36	0.17 ± 0.32	0.11 ± 0.26
P4	0.47 ± 0.45	0.29 ± 0.42	0.25 ± 0.40	0.18 ± 0.33	0.14 ± 0.31
PR4	0.61 ± 0.41	0.48 ± 0.44	0.42 ± 0.43	0.37 ± 0.43	0.26 ± 0.40
PR4CP1	0.70 ± 0.39	0.54 ± 0.43	0.49 ± 0.44	0.44 ± 0.45	0.29 ± 0.39
PR4CP2	0.48 ± 0.44	0.41 ± 0.44	0.38 ± 0.44	0.34 ± 0.44	0.26 ± 0.39
PR4ENV	0.75 ± 0.32	0.65 ± 0.39	0.61 ± 0.42	0.47 ± 0.44	0.34 ± 0.41
PR5ENV	0.80 ± 0.40	0.78 ± 0.39	0.71 ± 0.43	0.56 ± 0.47	0.39 ± 0.46

Table 5.4: MRR performance of monophonic queries

The results show that n-gram lengths of 3 are not sufficient for retrieval, whereas $n = 4$ brings about more satisfactory results (a MRR of 0.5 can be paraphrased as "typically, the known-item would be retrieved in second rank").

All indexing methods show a weakening with rising humming error levels in the query, but not in a drastic way. At least with $n \geq 4$, the retrieval performance is remarkably resilient even to high error rates of humming.

The combination of pitch and rhythm information seems beneficial over pitch alone. It has to be noted, though, that our humming error model does not include rhythm or timing. This is something we only investigated with polyphonic queries (see below).

Variations in coarseness with PR4, PR4CP1, PR4CP2 and selecting monophonic sequences to be indexed as with PR4ENV, do make a difference to the retrieval performance. This is caused by one of the most fundamental trade-offs in IR, the one between precision and recall. By indexing all possible combinations one is likely to increase recall but harm precision through a multitude of matched documents which are not relevant. Coarser encodings have a similar effect. However, for search engines and ad-hoc queries, users are more likely to be interested in high precision. This is why a reduction in the number of indexed musical words, i.e., selecting a number of musical words from all possible words is highly desirable, especially where reductions identify melodic lines, musical themes and other musically relevant melody fragments. It seems that the heuristics of keeping variations of the upper and lower envelopes with PR4ENV and PR5ENV helps remarkably well in terms of precision. Further analysis of restricting paths to improve retrieval performance is needed. This will have to include comparative studies of retrieval performances with PR4AM as discussed in Section 6.3.

Polyphonic Queries

Table 5.5 shows the mean reciprocal rank for polyphonic queries emulating the query-by-example case. Here the corresponding error model includes a model for rhythmic and timing errors. This table reinforces the analysis for monophonic queries. Again, $n \geq 4$ produces good results and, again, the methods are relatively robust under query errors. The heuristics of just keeping upper and lower envelopes proves to be a mechanism to boost precision. However, it needs to be noted that the fault-tolerance for PR4ENV under erroneous conditions is marginally better than PR5ENV. Again, more extensive tests would be required to investigate this fault-tolerance. Studies to compare retrieval performances with PR4AM for path restrictions would also be needed.

5.1.4 Experiment 4

This experiment was a preliminary investigation on proximity analysis towards improving retrieval performance. The aim was to test the feasibility of using 'overlaying' position in-

		$D_i = 1$	$D_i = 2$
Index	perfect query	$D_r = 0.02$	$D_r = 0.02$
P3	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PR3	0.35 ± 0.45	0.05 ± 0.11	0.04 ± 0.10
PR3CP1CR	0.11 ± 0.30	0.01 ± 0.04	0.01 ± 0.03
PR3CP2CR	0.10 ± 0.30	0.00 ± 0.01	0.00 ± 0.01
PR3ENV	0.62 ± 0.47	0.13 ± 0.31	0.05 ± 0.16
P4	0.76 ± 0.38	0.27 ± 0.38	0.10 ± 0.27
PR4	0.85 ± 0.31	0.36 ± 0.39	0.25 ± 0.38
PR4CP1CR	0.81 ± 0.38	0.23 ± 0.35	0.14 ± 0.30
PR4CP2CR	0.60 ± 0.49	0.16 ± 0.32	0.12 ± 0.30
PR4ENV	1.00 ± 0.00	0.57 ± 0.45	0.33 ± 0.43
PR5ENV	1.00 ± 0.00	0.56 ± 0.47	0.27 ± 0.42

Table 5.5: MRR performance of polyphonic queries

formation. The 'parser' of Lemur version 1.9 was enhanced. As predicted and discussed in Section 3.6, the ODN when tested with several known-item searches was unsuccessful. More complex query formulation, to test the significance of sequentiality as discussed in Section 4.1.4, was then performed. The SUM and ODN operators were used in combination forming nested phrase operators.

Four musical words formats were used for the study: PR4, PR4ENV, PPR4 and PPR4ENV. Two approaches to query formulation were compared, query with a 'bag of terms' and structured queries. With the 'bag of terms', weighting approaches of terms based on the vector-space model with the Okapi tf variant, similar to the previous two experiments were adopted. For the structured query formulation, the AND and the OR operators that enable retrieval with more relaxed conditions for term similarity were investigated. BAND and ODN that retrieve with strict retrieval conditions for term similarity were not used.

From Table 5.6, the highest MMR measure obtained, 0.70, clearly shows the feasibility of using proximity restrictions in musical document retrieval. It can also be noted that this improvement in retrieval performance was with full music indexing compared to the restriction of using envelopes, PPR4ENV. Loss of information, in this case, position information of musical words from monophonic sequences of 'middle voices' of the polyphonic structure,

Bag of terms	PR4	PR4ENV
	0.63	0.66
Structured	PPR4	PPR4ENV
SUM	0.57	0.62
Nested	0.70	0.66
AND	0.56	0.65
OR	0.61	0.62

Table 5.6: MRR Measures for 'bag of terms' and 'overlaying' words

affected the proximity based retrieval. With strict ordering more terms in the correct order needed to be found, as with the PPR4 format which enabled better retrieval than the 'bag of terms' approach of PR4. For all other query formulation, in general, a) the bag of terms approach, using the vector space model and the Okapi tf variant for term weights retrieved better compared to the use of any of the basic structured retrieval approaches, and b) reduced number of terms PR4ENV performed better. The robustness of the n-gram approach and the vector space model is once again shown. The scoring function proposed in Section 3.6 using notion of 'fuzzy match points' defined with music retrieval would need to be investigated for its usefulness as 'term frequencies'. Further investigations could look into ways of incorporating the frequency information into the vector space model, as this has been very successful with musical words(Doraisamy and Rüger 2003a; Zhai 2001).

5.2 Results Discussion

This section summarises the findings based on the main experimental factors listed in Chapter 4. Questions investigated for each factor are restated, followed by the findings discussion.

- Dimension: *Would the addition of the rhythm dimension to the information content of the 'musical' n-gram improve retrieval performance?* Yes, based on the better retrieval performances for PR4 compared to P4, it is clear from Experiment 1 through 3 that the inclusion of rhythm is an important addition. This extension to the information content of n-gram which has been a limitation to the study by Downie (1999) has been addressed, and proven to be successful and necessary.

- N-gram length: *Would the length of the n-gram representations affect performance?* Based on the results of the experiments performed, there is no clear evidence to recommend the value of n for the n-gram construction from polyphonic music data. Experiments 1 and 2 compared performances with values of $n = 3$ and 4. Performances with $n = 4$ were better — P4, PR4 performed better than P3, PR3 correspondingly. In Experiment 3, the value n=5 was investigated. Due to the large number of index terms generated this was tested with a reduced number of monophonic paths, from all possible paths, and performed well. However, it has to be noted that with polyphonic queries and this path restriction, PR4ENV performed marginally better than PR5ENV under error conditions. More experiments would be needed to compare the retrieval performances using $n = 4$ and 5.

- Encoding precision: *Would coarser encodings improve fault-tolerance?* Similar to the findings of Downie (1999), the expected fault-tolerance through the use of interval classes for encoding did not appear to exist. Experiment 1 appeared to show fault-tolerance although results of experiments 2 and 3 did not show this. The encoding precision does have an effect, there is better performance in some cases with PR4CP1 than PR4, such as in Experiment 1 for erroneous queries. However, this did not seem the case with Experiments 2 and 3. This fault-tolerance cannot be concluded without further investigation.

- Paths: *Would the selection of a number of paths compared to using all possible paths affect the retrieval performance?* Approaches to reduce the number of paths from all possible paths in full polyphonic music data without too much loss of information were looked into. The restriction of paths to the upper and lower envelopes performed well in Experiment 3, with improved performance shown by PR4ENV in comparison to PR4. The comparative study of Experiment 2 also shows that the approach motivated by Uitdenbogerd and Zobel (1999) on selecting only one path performed best. The restriction of paths to the envelopes was proposed on the heuristics that with more information the performance would improve. The results show that a simpler restriction to the highest pitch at each onset is better, at least for this data set. This restriction has to be investigated further.

- Contiguity: *Would skipping a number of contiguous onsets overcome the problem of intersecting accompanying onsets, that may pose a problem when querying a monophonic query against a polyphonically indexed collection?* No, it did not solve the problem. This was investigated in Experiment 2 and the performance degraded. However, again, PR4AM should be investigated as it performed for the retrieval using monophonic queries. Alternatively, skipping more onsets could be investigated, as only the skipping of one and two onsets were investigated.

- Position: *Would the incorporation of position information to the index information improve retrieval precision?* Experiment 4 indicates PPR4 with Nested phrase operators and strict ordering of terms improved the performance. However, the improvement was marginal. It was discussed in Section 3.6 that the scoring approaches of these operators may not be suitable since only adjacency is considered and not concurrency. This would have to be addressed with polyphonic music. A new proximity-based operator and scoring function was proposed which requires more in-depth evaluation.

5.3 Summary

Experimental results performed in four stages clearly show the successful use of our n-gram approach for polyphonic music retrieval. The results presented for polyphonic retrieval are very promising for queries in both polyphonic and monophonic form. These results extend qualitatively to queries, both humming and raw audio transcriptions, made under simulated error conditions and prove the robustness of the n-gram method. Experimental data also showed the need for further investigation on PR4AM as it appears particularly promising, in particular for monophonic queries.

Chapter 6

Conclusions and Discussions

Discussions on the contributions, limitations and the future directions of this work are presented. The future work discussion includes a system design for a polyphonic retrieval system that integrates music-friendly input and output (I/O) interfaces. An early polyphonic music retrieval system prototype is briefly discussed. In addition to this, the last section emphasizes the issue of a standardised testbed that is actively being addressed via the ongoing MIR/MDL Evaluation project. This includes a discussion of limitations and recommendations that was put forward to the MIR community in the 3rd edition of the MIR/MDL Evaluation project's white paper collection.

6.1 Contributions

This in-depth study has defined n-grams for polyphonic music retrieval and proven their usefulness. An novel interval mapping function was utilised to model the distribution of occurring intervals; this was shown to be valuable as well as the onset time ratios for incorporating rhythm information. The rhythm quantisation problem has been overcome by using bin widths that reflect the distribution of relevant and frequent ratios in music performance data. The results presented so far for polyphonic retrieval are qualitatively comparable to published successful monophonic retrieval experiments, e.g. by Downie (1999). Hence, they are very promising.

The salient contributions include:

- Defined a full-music indexing approach for polyphonic music
  - Introduced and evaluated a method for full-music indexing of polyphonic music, similar to full-text indexing or full-music indexing that has only been previously performed with monophonic sequences.
- Extended the musical n-gram information content
  - Shown that the information content of a musical n-gram can be extended to include rhythmic data. Only the pitch dimension has been used previously and the addition of rhythmic information has been shown to improve retrieval performance.
- Proposed a data-driven encoding approach
  - Analysed interval and rhythmic ratio distributions of the data-set; this led to a data-driven encoding of musical words from musical n-grams.
  - Introduced a parameter to the novel interval-mapping function that allows for different coarseness of the information encoding.
  - Identified individual ratio bin widths for the encoding of rhythmic data; they too enable variations to the coarseness of rhythmic information encoding.
- Proposed ways to tackle problems using n-grams for polyphonic music
  - Hypothesised that alternate onsets could overcome the problem of intercepting onsets from a polyphonic document when queried with a monophonic melody (hypothesis not proven).
  - Restricted all possible paths to envelopes to overcome problems of a large number of possible index terms.
  - 'Overlayed' word positions to enable proximity-based retrieval for adjacent and concurrent musical words.

- Utilised text search engines and IR models for music retrieval
  - Used text search engines with polyphonic musical words; existing variants to the vector space model have been tested.
  - Incorporated 'overlaying' position information into an existing search engine to define a new proximity-based ranking operator; this operator uses 'fuzzy match points'.
- Outlined an experimental framework for MIR
  - Developed a small-scale test collection that can continue to be used by other researchers.
  - Surveyed and adopted errors models for the fault-tolerance study within this framework; this enabled the robustness of the n-gram method to be shown.
- Analysed findings and provided a list of recommendations

The work in this thesis has contributed the following primary literature:

- S Doraisamy and S Rüger (2001), An Approach Towards a Polyphonic Music Retrieval System, In Proc. of the Second International Symposium on Music Information Retrieval, ISMIR 2001, Indiana, USA, pp 187–193.
- S Doraisamy and S Rüger (2002), A Comparative and Fault-tolerance Study of the Use of N-grams with Polyphonic Music, In Proc. of the Third International Symposium on Music Information Retrieval, ISMIR 2002, Paris, France, pp 101–106.
- S Doraisamy and S Rüger (2003), Robust Polyphonic Music Retrieval with N-grams, Journal of Intelligent Information Systems, Volume 21, Number 1, July 2003, Kluwer Academic Publishers, pp 53–70.
- S Doraisamy and S Rüger (2003), Emphasizing the Need for TREC-like Collaboration Towards MIR Evaluation, presented at the Workshop on the Evaluation of Music Information Retrieval (MIR) Systems, SIGIR 2003, Toronto, Canada, and published in The MIR/MDL Evaluation Project White Paper Collection, Edition #3, pp 90–96.

- S Doraisamy and S Rüger (2003), Position Indexing of Adjacent and Concurrent N-grams for Polyphonic Music Retrieval, in Proc. of the Fourth International Conference on Music Information Retrieval, ISMIR 2003, Maryland, USA, pp 227–228.

- S Doraisamy and S Rüger (2004), A Polyphonic Music Retrieval System using N-grams, in Proc. of the Fifth International Conference on Music Information Retrieval, ISMIR 2004, Barcelona, Spain, in print.

6.2 Limitations

- Error models

  - Further study would be required for rhythmic error models. Error models not just on QBH but also aural and perception studies would need to be incorporated.
  - Coarser encodings for rhythm and pitch would have to be investigated further.

- Large-scale test collection, see also Subsection 6.4

  - The test collection developed used a largely classical collection. Equal numbers of music performances from a number of genres should be included.
  - The relevance judgements assumption would need to include more expert opinions.
  - Real-life queries should be included to enable a more comprehensive evaluation of our approach.

- Our encoding strategy has addressed Western musical notation. The data-driven encoding approach should enable this model to be easily adopted for any music notation, such as using a combination of two text characters to encode intervals from different musical scales.

6.3 Future Work

- Proximity based retrieval: The feasibility of incorporating 'overlaying' word position has been shown with a preliminary test. The scoring function for the proposed MODN

operator has been tested with several known-item searches. This requires further evaluation with the possibility of combining scoring approaches with the Okapi tf method that was shown to be robust.

- Restriction of paths: Further investigation comparing performances of path restriction based on envelopes, and the all-mono approach would be needed. Results indicate that PR4AM is promising. However, the loss of information when indexing would have be evaluated with a larger set of both monophonic and polyphonic queries.

- Information content: An investigation should be performed on the possibility of extending the information content of a musical n-gram further to include other dimensions of music, e.g., timbre.

- Sample size: A small query sample has been used with most of the experimentation. The test collection with the larger query set and extensive relevant document list was developed towards the end of this study and this would be utilised for further tests.

- IR models: The main IR model adopted for this study has been the vector space model. Although our initial investigations on the use on the probabilistic approach such as language models was not promising, the development of statistical models for musical word sequences for a particular genre or works of a particular composer could be investigated.

- Polyphonic music retrieval system design: Based on the MIR system overview that was shown in Figure 1.3, a polyphonic music retrieval system incorporating music-friendly I/O interfaces is shown in Figure 6.1.

  The screen shots of our early polyphonic music retrieval system prototype are shown in Figure 6.2. The input modes include a 'graphical keyboard' (obtained from the Java sound library) enabling monophonic performance input. The mouse event will generate a MIDI file which then will be processed by the musical document generator discussed Section 3.4. The query will be processed, generating query terms in the same format as the indexed documents for the search and retrieval. Indexing word formats were listed in Table 4.2. The screen shot shows retrieval results as a text file with a ranked list of documents. The 'Happy Birthday Song' was input via the graphical keyboard and it

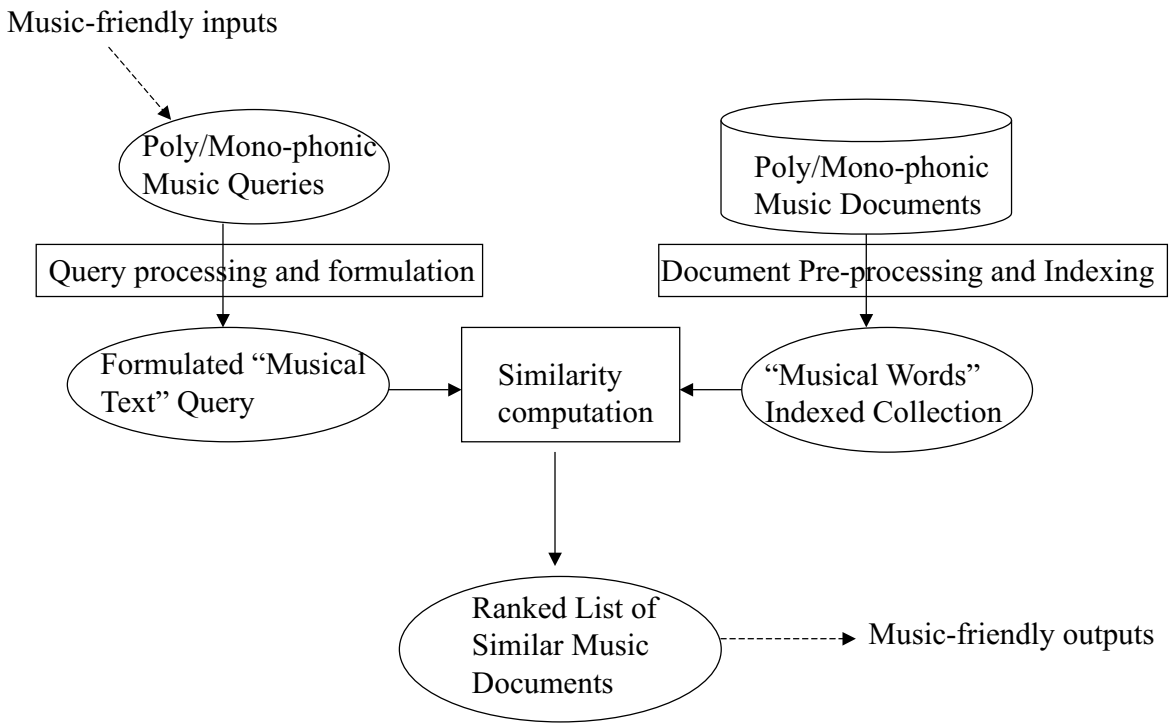


Figure 6.1: Polyphonic music retrieval system overview

can be seen that the song title 'birthday.mid' was retrieved at rank 1. For the retrieval shown, the musical text documents were indexed with the PR4ENV format with Lemur version 1.9. The system currently uses Lemur Version 2.2 that was enhanced for the inclusion of the new musical words parser, MODN and its scoring module.

The work in this thesis is part of the Multimedia Information Retieval research group's framework for content-based information retrieval. With our musical n-grams indexed as text words, music-friendly inputs such as query-by-humming and query-by-example that could be a polyphonic audio recording, will require more extensive query pre-processing modules. We worked with two Master level students who worked with pitch tracking and polyphonic transcription pre-processing modules. Figure 6.3 shows the QBH interface developed by Tarter (2003) where one can hum a query which would be transcribed by a pitch tracker written by him. This query would then be converted to the PR4ENV format for search and retrieval of musical words indexed using Lemur version 1.9. This format was adopted for the early prototype as well.

Other interfaces to music inputs that have been developed include a text contour input

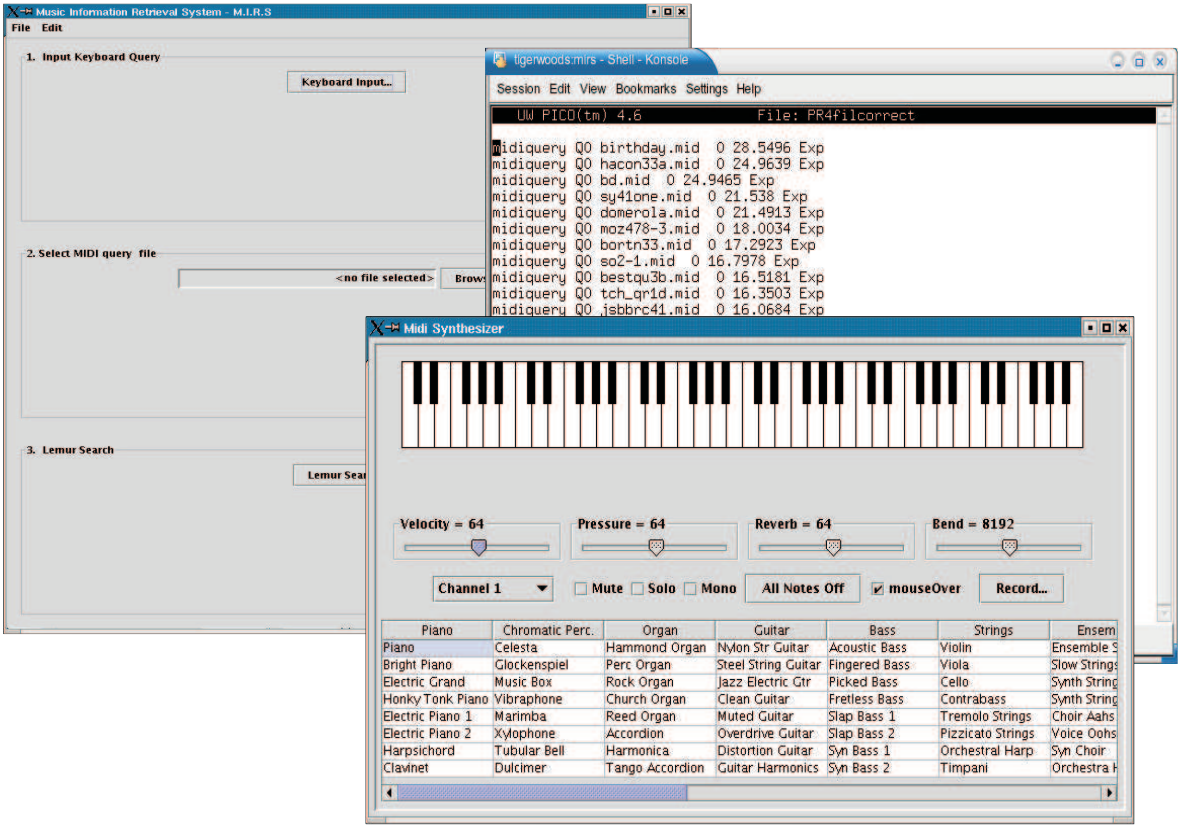


Figure 6.2: Early prototype

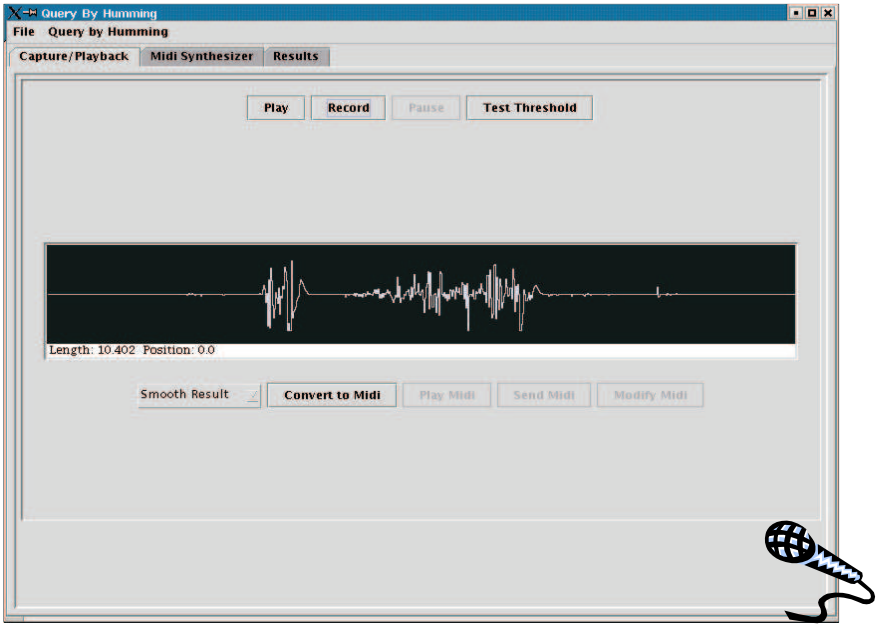


Figure 6.3: QBH system interface

by Walters (2001) and polyphonic audio file input, a Masters group project by Kay et al (2001). The latter used a polyphonic transcription algorithm developed by von Schroeter (2000) for the input pre-processing module. Screen shots of these are shown in Figure 6.4.

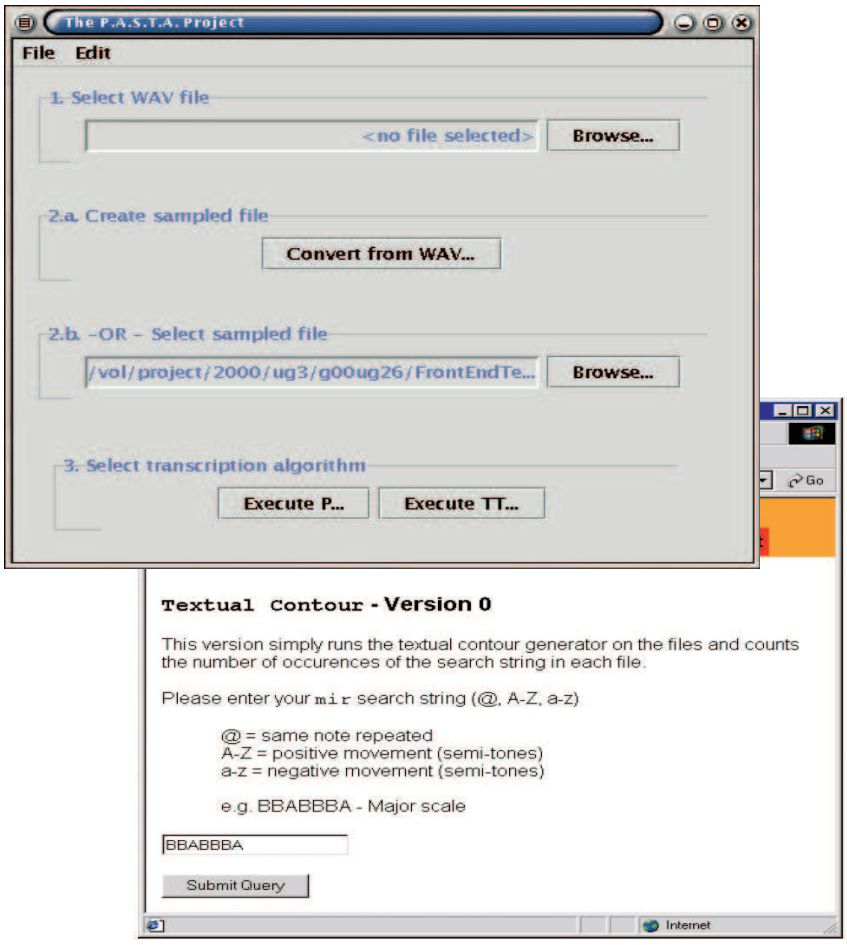


Figure 6.4: Monophonic text contour and polyphonic audio inputs

Further work plans include work on music-friendly outputs. This would be auralisation, analogous to visualisation. One possible approach would be that for a piece retrieved with rank 1, a media player would automatically be launched to play the first few seconds of the music file. More sophisticated approaches from Human Computer Interaction (HCI) studies for visualisation of ranked list of documents could be incorporated. Finally, for the web deployment of the early prototype, Lemur 2.2 includes CGI scripts for this.

6.4 Standardised Testbed

The review of MIR studies presented in Chapter 2 shows that a number of researchers have developed individual state-of-the-art test collections for the purpose of metric scientific evaluation. However, there are a number of potential problems that are generic to these various small-scale test collections. These include the lack of consistency and the incompleteness of the relevance judgements; this discussion is presented in the following subsection. Standard evaluation measures (modified in some cases) have been shown to be useful for MIR evaluation. Whether or not such measures are useful for large-scale evaluation in MIR context is a question that needs to be investigated further. The last subsection emphasizes that many of the limitations generic to all these studies could be overcome through the development of standardised large-scale test collections.

6.4.1 Generic Problems

Standardised test collections have been a useful benchmark for text retrieval evaluation for around 40 years now (Voorhees 2002). Manual relevance judgements, the pooling approach, known-item searches, relevance by song titles/filenames and exhaustive judging have all been used with MIR test collections. All of these approaches, however, have potential problems such as completeness and consistency, defined in Section 4.1, of the relevance judgements. These problems have been addressed with the large TREC collections and will need to be addressed when moving away from the small-scale test collections towards large-scale evaluation.

For a test collection, it has been said that what has to be addressed is not so much how well assessors agree with one another, but how evaluation results change with differences in assessments (Voorhees 2000). The stability of the test collection is an important issue and has been addressed by TREC. Relative effectiveness of two retrieval strategies should be insensitive to slight changes in the relevant document set in order to reflect the true merit of the retrieval strategy being evaluated. The reasons given by Lesk and Salton (1969) for stability of system rankings despite differences in relevance judgements have been further discussed by Voorhees (2000). They are as follows:

- Evaluation results are reported as averages over many topics.
- Disagreements among judges affect borderline documents, which are usually ranked

after documents that are unanimously agreed upon.

- Recall and precision depend on the relative position of the relevant and non-relevant documents in the relevance ranking, and changes in the composition of the judgement sets may have only a small effect on the ordering as a whole.

It has been argued that the third reason may not apply to the large collection sizes of TREC where there could be hundreds of relevant documents. It has been shown, however, that first and the second reason appear to hold for the TREC collections (Voorhees 2000). In the context of MIR, it is unclear to what extent the second reason will prove valid, for it could be argued that human music perception would generate much larger differences in relevance judgements. This question clearly needs to be investigated further.

The investigation into the stability of system rankings with different sets of relevance assessments was carried out by NIST with the following tests as described in Voorhees (2000):

- The use of the overlap of the relevant document sets to quantify the amount of agreement among different sets of relevance assessments (Lesk and Salton 1969). Overlap is defined as the size of intersection of the relevant document sets divided by the size of the union of the relevant document sets.

- As a different view of how well assessors agree with one another, one set of judgements, say set Y, can be evaluated with respect to another set of judgements, set X. Assume the documents judged relevant in set Y are the retrieved set; then the recall and precision of that retrieved set using the judgements in X can be calculated.

- The correlation can be quantified by using a measure of association between the different system rankings. A correlation based on Kendall's τ as the measure of association between two rankings can be used. Kendall's τ computes the distance between two rankings as he minimum number of pair-wise adjacent swaps to turn one ranking into the other. The distance is normalised by the number of items being ranked such that two identical rankings produce a correlation of 1.0, the correlation between a ranking and its perfect inverse is -1.0, and the expected correlation of two rankings chosen at random is 0.0.

When looking at completeness, it is also necessary to assess the degree of selection bias that occurs. Relevance judgements need to be unbiased, i.e., it does not matter how many or how few judgements are made, but the documents that are judged should not be correlated with the documents in a particular retrieval method. Having complete judgements ensures that there is no selection bias, but pooling with sufficiently diverse pools has been shown to be a good approximation (Voorhees 2000).

TREC-like collaboration is clearly needed to conduct the extensive tests required to address stability issues of MIR test collections and to obtain sufficiently diverse pools. Such tasks as described above are formidably difficult for individual researchers to accomplish at a smaller scale.

6.4.2 TREC-like Collaboration

We believe that potential problems that are generic to the various small-scale test collections discussed can be overcome through TREC-like collaboration. This could alleviate problems in the following areas:

- Resources (Collection and queries): The collection sizes used are a fraction of real-word music repositories (Downie 2003b) and much larger collection sizes are needed. The difficulties in query acquisition may be overcome by obtaining real-world queries such as those available with music libraries, radio stations, scanning or encoding documented themes. Extensive studies on real-world error models of music queries would be one way towards generating a large music query repository. A collaborative effort would be needed to identify potentially relevant documents (via pooling) and deliver more resources to assess the relevance of the pooled documents. A test bed created in this manner and made available would further the whole research field of Music IR.

- Relevance: Perhaps within the QBM task, relevance assumptions such as those already used in the studies thus far could be expanded. A comprehensive representation of user classes may be needed to reach a consensus on the relevance based on a task. Various tasks need to be identified and the relevance defined accordingly. Relevance based on a scale is one possibility as human musical perception of similarity is notoriously difficult to model.

- Copyright: TREC mainly uses old newspaper articles, which have no real commercial value, and hence distribution is not problematic. Music pieces have a value, and the test bed needs to be protected either through drastic licenses specifying the legal use of the data or perhaps by storing the test collection in a centralised secure environment; this in turn could offer computing services such as the downloading of search engine indexing and retrieval code that is then executed at the repository and receives the ranked lists. Alternatively, one could get preprocessed "features" of music pieces which are commercially not relevant (as opposed to the original music piece from the audio): a midi-like representation, the volume footprint, a rhythm or pitch extract, etc. Yet another way could be a test collection thats consists of "half" music pieces, the first 60 seconds of each 120 seconds segment of the music piece, etc.

- Generic modules: Collaboration does not have to end with the creation of test collection. A collaborative effort to develop modules that create features, extract melodies, preprocess data would certainly benefit MIR research tasks. For example, groups with an expertise in IR of symbolic representation might well benefit from preprocessing that translates raw audio into symbolic forms such as MIDI.

Appendix A

Query-Relevance Set

Relevant_num.qrels file

01 0 allaturk.mid 1

01 0 rondturc.mid 1

01 0 turca.mid 1

01 0 turcmoz.mid 1

01 0 turk_ron.mid 1

02 0 3birth.mid 1

02 0 99.mid 1

02 0 bd.mid 1

02 0 birthday.mid 1

03 0 CHARIOT2.MID 1

03 0 chariots.mid 1

03 0 chariots3.mid 1

04 0 ch-etn03.mid 1

05 0 eineklei.mid 1

05 0 eknm1.mid 1

05 0 kslegro.mid 1

05 0 nachtmsk.mid 1

05 0 night-3.mid 1

06 0 5beet1mv.mid 1

06 0 beet5m1.mid 1

06 0 besy51.mid 1

06 0 besym5-1.mid 1

06 0 classica.mid 1

06 0 fifth_sy.mid 1

06 0 gmb5m1.mid 1

06 0 lvbsym51.mid 1

07 0 bwv846.mid 1

07 0 wtc1012.mid 1

08 0 be-elise.mid 1

08 0 furelise.mid 1

08 0 furelse1.mid 1

09 0 GARDEN.MID 1

09 0 cgardens.mid 1

10 0 alleluia.mid 1

10 0 conta10.mid 1

10 0 hallelu.mid 1

10 0 hallujah.mid 1

10 0 hanme42.mid 1

10 0 mesiah44.mid 1

10 0 n44.mid 1

Appendix B

Test Collection: Query and Relevant Document List

No.	Composer	Query	Relevant Document
1	J.S. Bach	Air for the G String (QJSBOS32.MID)	jsbos32.mid*
			bachair.mid
			gp_air_g.mid
			sui3_2ai.mid
			bjsair.mid
2		Toccata and Fugue in D Minor, BWV 565 (Q565_TOC.MID)	565_tocf.mid*
			bach-565.mid
			tocba.mid
3		Brandenburg Concerto, No. 2, 1st Mvt (Q1BRAN2-.MID)	lbran2-1.mid*
			bjs10471.mid
			jsbbrc21.mid
4		Jesu, Joy of Man's Desiring (QJESU_J1.MID)	jesu_j1.mid*

*Indicates the file the monophonic query was manually extracted from. Theme extraction was based on references to the Dictionary of Musical Themes (Barlow and Morgenstern 1949), and if unavailable, references were made to sheet music downloaded from www.music-scores.com.

/..cont.

No.	Composer	Query	Relevant Document
5	Beethoven	Moonlight Sonata (QMOONL1.MID)	moonl1.mid*
			moonlit.mid
			moonltel.mid
			besn14-1.mid
			alchiadi.mid
			be-ps-14.mid
			moonlite.mid
6	Brahms	Hungarian Dance No. 5 (QBR-HD-0.MID)	br-hd-05.mid*
			hungdnc5.mid
			hungdan5.mid
			brm_hgd5.mid
7	Chopin	Nocturne no,2 in E flat, opus 9 (QNOC4.MID)	noc4.mid*
8		Waltz in E flat Major, Opus 18 (QCHOVALO.MID)	vals1_eb.mid
			choval01.mid*
			choval06.mid
			w64-1.mid
			ch_w64_1.mid
			chopineb.mid
9		Waltz, Opus 64, No. 2 in C sharp Minor (WALTZ.MID)	choval07.mid*
			chop64_2.mid
			chop642.mid
			chwl64-2.mid
			cshrpvls.mid
10		Waltz No. 2 in B Minor, opus 69 (WALTZ2B.MID)	chpval10.mid*
			chvai7pb.mid

/..cont.

No.	Composer	Query	Relevant Document
11	Debussy	Golliwog's Cakewalk(QDEBGOLL.MID)	debgolli.mid*
12	Haydn	Emperor's Quartet, 2nd Mvt (QEMPVAR.MID)	empvar.mid*
13		Surprise Symphony, 2nd Mvt(QHAYDN942.MID)	haydn942.mid*
14	Liszt	Liebestraum no. 3 (QLIEBSTR.MID)	liebstrm.mid*
15	Mendelssohn	Midsummer Night's Dream (Wedding March) (QMNSTO9.MID)	mnsnt09.mid*
			fmbwedmc.mid
			mnsnt12.mid
			mendwd1.mid
			brasswed.mid
			hochzeit.mid
			m_wmarch.mid
			wedding.mid
			bridal.mid
			bridalch.mid
16		Symphony no. 4, 1st mvt (QITALIAN.MID)	italian1.mid*
17	Mozart	Variations in C, Theme: "Ah, Vous Dirai-Je, Maman" (QK165.MID)	k165.mid
			wamk265.mid
			sda_tav.mid
18		Symphony No. 40, 1st mvt (QK550_1.MID)	k550_1.mid
			sym40-1.mid
			sym41gm.mid
			mozarts2.mid
19	Schubert	Schwanengesang : Ständchen (Serenade) (QSCHRSND.MID)	schsrnd.mid*
			standche.mid

/..cont.

No.	Composer	Query	Relevant Document
20	Schumann	Kinderscenen — No. 1 (QKDRZN_0.MID)	kdrzn_01.mid*
			schkinhp.mid
21	Tchaikovsky	Piano Concerto no.1, 1st mvt (QTCHAI1-.MID)	tchai1-1.mid*
			tch-pcon-1.mid
22		Swan Lake (Intro) (QSWANLK-.MID)	swanlk-1.mid*
23	Dvorak	Slavonic Dances, No. 1 (QSLAVDNC.MID)	slvdnce.mid*
			dvn1o46.mid
			sltan11.mid
24		Slavonic Dances, No. 8 (QGP_SLAV.MID)	gp_slav8.mid*
			dvn8o46.mid
			sltan18.mid
25		Serenade for Strings, 1st mvt (QDVOP22-.MID)	dvop22-1.mid*
26		Serenade for Strings, 2nd mvt (QSERENAD.MID)	dvop22-2.mid*
27	Delibes	Valse Lente (Coppelia) (QV_LENTE.MID)	v_lente.mid
28	Bizet	Carmen (Prelude to Act 1) (QBIZCART.MID)	bizcarto.mid*
			bzsmcrmn.mid
29		Carmen (Habenera) (QHABAN.MID)	haban.mid
30	Elgar	Pomp and Circumstance (QPOMPCIR.MID)	pompcirc.mid
31	Faure	Dolly Suite (Berceuse) (QDOLL.MID)	gp_dolly.mid*
			dolly1.mid
32		Sicilenne (QF_SICIL.MID)	f_sicili.mid
			18sicily.mid
33	Ravel	Pavane for the Dead Princess (QPAVINF.MID)	pavinf.mid*
			ravpavan.mid
			diana.mid
			cippuid.mid
			ravelpav.mid

/..cont.

No.	Composer	Query	Relevant Document
34	Grieg	Peer Gynt Suite (Morning Mood) (QPG1MORN.MID)	pg1morn.mid*
			gr-peel1.mid
			morn.mid
			ggynt.mid
			1_morng.mid
35		Peer Gynt Suite (Hall of the Mountain King) (QIN_THE_.MID)	in_the_h.mid*
			gp_hall.mid
			peergyn4.mid
			pg1king.mid
			4_mtking.mid
36	Mussorgsky	Pictures at an Exhibition (Promenade) (QPROMPIX.MID)	prompix.mid*
			pm1gnome.mid
			pmarkcat.mid
			pmcastle.mid
			pmtuiler.mid
			pxone.mid
			prom2.mid
			bilder2.mid
			bilder3.mid
			promin-1.mid
37	Scott Joplin	Maple Leaf Rag (QMLRAG.MID)	mlrag.mid*
			sj_mlrjf.mid
			mleafrag.mid
38		Entertainer (QENTERTA.MID)	entertai.mid*
			jopenter.mid

/..cont.

No.	Composer	Query	Relevant Document
39	Smetana	Moldau (1st theme) (QVLTAVA.MID)	vltava.mid*
			moldvlat.mid
40	Strauss	Radetzky March (QRADETZK.MID)	radetzky.mid*

Appendix C

Retrieval Performance Measures from Experiment 3

Detailed retrieval performance measures for each of the 10 queries listed in Table 4.3 (in Chapter 4) are shown in Tables C.1, C.2 and C.3. Weighted averages listed in the last row of each table are summariesd and shown in Table 5.3 (in Chapter 5).

Song ID	PR4	PR4CP1	PR4AL1	PR4AL2	PR4AM	P4
1	100	0	0	0	100	0
2	50	25	50	25	50	25
3	0	0	0	0	0	0
4	0	0	0	0	100	0
5	100	40	100	100	100	0
6	13	0	0	0	100	0
7	100	50	100	50	100	0
8	33	33	33	0	67	67
9	50	50	50	50	50	0
10	86	14	71	43	86	0
W.A.	58	18	40	34	80	8

Table C.1: Percentage of relevant documents retrieved within rank 15 with perfect queries

Song ID	PR4	PR4CP1	PR4AL1	PR4AL2	PR4AM	P4
1	10	0	0	4	60	0
2	40	5	43	16	50	10
3	0	0	0	0	0	0
4	0	0	0	20	100	0
5	92	28	90	86	92	0
6	10	0	10	0	89	0
7	85	45	85	45	90	0
8	30	30	26	0	47	27
9	50	40	50	40	50	0
10	86	19	71	40	86	0
W.A.	43	14	39	25	70	3

Table C.2: Percentage of relevant documents retrieved within rank 15 with error probability of 10%

Song ID	PR4	PR4CP1	PR4AL1	PR4AL2	PR4AM	P4
1	0	0	0	0	2	0
2	28	3	23	5	50	0
3	0	0	0	0	0	0
4	0	0	0	0	90	0
5	82	24	84	9	90	0
6	9	0	0	0	78	0
7	70	15	75	35	70	0
8	30	23	7	0	40	0
9	50	40	50	50	50	0
10	86	9	71	26	82	0
W.A.	38	9	32	10	58	0

Table C.3: Percentage of relevant documents retrieved within rank 15 with error probability of 20%

Bibliography

E Adams (1991). *A Study of Trigrams and their feasibility as Index terms in a full text Information Retrieval System*. PhD thesis, George Washington University.

J Allan, M E Connell, W B Croft, F Feng, D Fisher and X Li (2000). Inquery at TREC-9. In *The Ninth Text REtrieval Conference (TREC-9)*, pp 551—562.

R Baeza-Yates and B Ribeiro-Neto (1999). *Modern Information Retrieval*. ACM Press Addison Wesley.

D Bainbridge, C G Nevill-Manning, I H Witten, L A Smith and R J McNab (1999). Towards a digital library of popular music. In *Fourth ACM Conference on Digital Libraries, DL '99*, pp 161–169.

H Barlow and S Morgenstern (1949). *A Dictionary of Musical Themes*. London: Ernest Benn.

J P Bello and M Sandler (2003, April). Phase-based note onset detection for music signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-03)*, Hong Kong.

Theodore Besterman (1945). Introductory note. *The Journal of Documentation 1*(1), 1.

S Blackburn and D DeRoure (1998). A tool for content-based navigation of music. In *ACM Multimedia '98*, pp 361–368.

A Bonardi (2000). IR for contemporary music: What the musicologist needs. In *1st International Symposium on Music Information Retrieval, ISMIR 2000*.

J Broglio, J P Callan, W B Croft and D W Nachbar (1994, November). Document retrieval and routing using the Inquery system. In *NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3)*, Gaithersburg,Maryland, pp 29–38.

D Byrd (2000). Candidate music test collections. In *1st International Symposium on Music Information Retrieval, ISMIR 2000*.

D Byrd (2001). Nightingale. http://www.ngale.com. Adept Music Notation Solutions.

D Byrd and T Crawford (2002). Problems of music information retrieval in the real world. *Information Processing and Management 38*, 249–272.

J P Callan (1994). Passage-level evidence in document retrieval. In *Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 94*, Dublin, Ireland, pp 302–310.

J P Callan, W B Croft and S M Harding (1992). The Inquery retrieval system. In *Third International Conference on Database and Expert Systems Applications*, Valencia, Spain, pp 78–83. Springer-Verlag.

E Cambouropoulos, T Crawford and S I Costas (1999). Pattern processing in melodic sequences: Challenges, caveats & prospects. In *AISB'99 Symposium on Musical Creativity*, Edinburgh, pp 42–47.

A L P Chen, M Chang, J Chen, J L Hsu, C H Hsu and S Y S Hsu (2000). Query by music segments: An efficient approach for song retrieval. In *IEEE International Conference on Multimedia and Expo, ICME 2000*, pp 873–876 vol 2.

J C C Chen and A L P Chen (1998). Query by rhythm: An approach for song retrieval in music databases. In *IEEE International Workshop on Research Issues in Data Engineering*, pp 139–146.

Y Chiaramella (2000, September). Information retrieval and structured documents. In M Agosti, F Crestani and G Pasi (Eds), *Lectures on Information Retrieval - Third European Summer-School, ESSIR 2000*, pp 286–309. Varenna, Italy: Springer-Verlag.

T C Chou, A L P Chen and C C Liu (1996). Music databases: Indexing techniques and implementation. In *IEEE International Workshop on Multimedia Database Management Systems*, pp 46–53.

S G Choudhury, T DiLauro, M Droettboom, I Fujinaga, B Harrington and K Macmillan (2000). Optical music recognition system within a large-scale digitisation project. In *1st International Symposium on Music Information Retrieval, ISMIR 2000*.

M Clausen, R Engelbrecht, D Meyer and J Schmitz (2000). PROMS: A web-based tool for searching polyphonic music. In *1st International Symposium on Music Information Retrieval, ISMIR 2000*.

C Cleverdon (1967). The Cranfield tests on index language devices. *Aslib Proceedings 19*, 173–192. Reprinted in K Spärck Jones and P Willett (eds): Readings in Information Retrieval, Morgan Kaufmann Publishers, 1997.

C Cleverdon (1970, March). Evaluation tests of information retrieval systems. *Journal of Documentation 26*(1), 55–67.

C W Cleverdon (1991). The significance of the Cranfield tests on index languages. In *14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1991*, pp 3–12.

F M Comlekoglu (1990). *Optimising a Text Retrieval System Utilising N-gram Indexing*. PhD thesis, George Washington University.

D Cope (1998). Signatures and earmarks: Computer recognition of patterns in music. *Computing in Musicology 11*, 129–138.

T Crawford, C S Iliopoulus and R Raman (1998). String-matching techniques for musical similarity and melodic recognition. *Computing in Musicology 11*, 73–100.

M Crochemore, C S Iliopoulus, T Lecroq and Y J Pinzon (2001). Approximate string matching in musical sequences. In *Prague Stringology Club Workshop, PSCW'01*, pp 26–36.

B Croft, J Callan and J Broglio (1993, November). TREC-2 routing and ad-hoc retrieval evaluation using the Inquery system. In *NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2)*, Gaithersburg,Maryland, pp 75–84.

M Dillon and M Hunter (1982). Automated identification of melodic variants in folk music. *Computers and the Humanities 16*, 107–117.

S Dixon (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research 30*(1), 39–58.

S Doraisamy (1995). Locating recurring themes in musical sequences. Master's thesis, University Malaysia Sarawak.

S Doraisamy and S Rüger (2001). An approach towards a polyphonic music retrieval system. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 187–193.

S Doraisamy and S Rüger (2002). A comparative and fault-tolerance study of the use of n-grams with polyphonic music. In *3rd International Conference on Music Information Retrieval, ISMIR 2002*, pp 101–106.

S Doraisamy and S Rüger (2003a). Emphasizing the need for TREC-like collaboration towards MIR evaluation. Workshop on the Evaluation of Music Information Retrieval (MIR) Systems, SIGIR 2003, and published in the MIR/MDL Evaluation Project White Paper Collection, Edition # 3, pp 90–96.

S Doraisamy and S Rüger (2003b). Position indexing of adjacent and concurrent n-grams for polyphonic music retrieval. In *Fourth International Conference on Music Information Retrieval, ISMIR 2003*, pp 227–228.

S Doraisamy and S Rüger (2003c). Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems 21*(1), 53–70.

M Dovey (2001). A technique for regular expression style searching in polyphonic music. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 179–185.

S Downie (1999). *Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text*. PhD thesis, University of Western Ontario.

S Downie (2002a). The MIR/MDL evaluation project white paper collection edition #2. http://www.music-ir.org. Editor and Project Organiser.

S Downie (2002b). Report on ISMIR2002 conference panel I: Music information retrieval evaluation frameworks. *D-Lib Magazine 8*(11).

S Downie (2003a). The MIR/MDL evaluation project white paper collection edition #3. http://www.music-ir.org. Editor and Project Organiser.

S Downie (2003b). Music information retrieval. *Annual Review of Information Science and Technology 37*, 295–340.

S Downie and M Nelson (2000). Evaluation of a simple and effective music information retrieval method. In *SIGIR 2000*, pp 73–80.

J W Dunn and C A Mayer (1999). VARIATIONS: A digital music library system at Indiana University. In *Fourth ACM Conference on Digital Libraries*, pp 12–19.

A S Durey and M A Clements (2001). Melody spotting using Hidden Markov Models. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 109–117.

J Foote (1999). An overview of audio information retrieval. *Multimedia Systems* 7(1), 2–11.

W B Frakes (1992). *Information Retrieval: Data Structures and Algorithms*, Chapter Introduction to Information Storage and Retrieval Systems, pp 1–12. Prentice Hall.

G H Gonnet, R A Baeza-Yates and T Snider (1991). Lexicographical indices for text: Inverted files vs. PAT trees. Technical Report OED-91-01, Centre for the New OED and Text Research, University of Waterloo.

S M Harding, W B Croft and C Weir (1997). Probabilistic retrieval of OCR degraded text using n-grams. In *Research and Advanced Technology for Digital Libraries*, pp 345–359.

D Harman (1992). Ranking algorithms. In W B Frakes and R Baeza-Yates (Eds), *Information Retrieval: Data Structures and Algorithms*, pp 363–392. Prentice Hall.

S P Harter and C A Hert (1997). Evaluation of information retrieval systems: Approaches, issues and methods. *Annual Review of Information Science and Technology 32*, 1–94.

G Haus and E Pollastri (2001). An audio front end for query by humming systems. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001.*

H S Heaps (1978). *Information Retrieval: Computational and Theoretical Aspects.* Academic Press.

M A Hearst (1996, April). Improving full-text precision on short queries using simple constraints. In *Fifth Annual Symposium on Document Analysis and Information Retrieval, SDAIR '96*, Las Vegas, Nevada.

W Hewlett (2001). An electronic library of classical music scores. http://www.musedata.org. Centre for Computer Assisted Research in the Humanities.

W B Hewlett and E Selfridge-Field (1997). MIDI. In E Selfridge-Field (Ed), *Beyond MIDI: The Handbook of Musical Codes*, pp 41–79. The MIT Press.

L Hirshman (1998, May). Language understanding evaluations: Lessons learned from MUC and ATIS. In *First International Conference on Language Resources and Evaluation, (LREC)*, pp 117–122.

L Hoffman-Engl (2002). Report on ISMIR 2002 conference panel III: Similarity in music. *D-Lib Magazine 8*(11).

H H Hoos, Hamel K A, Renz K and J Killian (1998). The GUIDO notation format - a novel approach for adequately representing score-level music. In *ICMC '98*, pp 451–454.

H H Hoos, K Renz and M Görg (2001). GUIDO/MIR - an experimental musical information retrieval system based on GUIDO music notation. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 41–50.

J L Hsu, C C Liu and A L P Chen (1998). Efficient repeating pattern finding in music databases. In *Seventh International Conference on Information and Knowledge Management, CIKM '98*.

D Huron (1997). Humdrum and Kern: Selective feature encoding. In E Selfridge-Field (Ed), *Beyond MIDI: The Handbook of Musical Codes*, pp 375–401. The MIT Press.

D Huron (2000). Perceptual and cognitive applications in music information retrieval. In *1st International Symposium on Music Information Retrieval, ISMIR 2000*.

C S Iliopoulos, T Lecroq, L Mouchard and Y J Pinzon (2000). Computing approximate repetitions in musical sequences. In *Prague Stringology Club Workshop, PSCW'01*, pp 49–59.

R Kay, K Orbell, D Pandya, L Sirett and R Parmar (2001). Performance analysis suite for transcription algorithms on polyphonic music, the PASTA project. Technical report, Department of Computing, Imperial College London.

E M Keen (1991, April). The effectiveness of term position and frequency for output ranking. In T McEnery (Ed), *British Computer Society 13th Information Retrieval Colloquium*, University of Lancaster, pp 22–37.

A Kornstadt (1998). Themefinder: A web-based melodic search tool. *Computing in Musicology 11*, 231–236.

N Kosugi, Y Nishihara, T Sakata, M Yamamuro and K Kushima (2000). A practical query by humming system for a large music database. In *ACM Multimedia 2000*.

G Kowalski (2000). *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers.

J Kruskal (1999). An overview of sequence comparison. In D Sankoff and J Kruskal (Eds), *Time Warps, String Edits and Macromolecules*, pp 1–44. CSLI Publications.

J Lafferty and C Zhai (2001, September). Document language models, query models, and risk minimization for information retrieval. In *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001*, pp 111–119.

W Lee and A L P Chen (2000). Efficient multi-feature index structures for music data retrieval. In *SPIE Conference on Storage and retrieval for Image and Video Databases*.

K Lemström (2000). *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki.

K Lemström, A Haapaniemi and E Ukkonen (1998). Retrieving music — to index or not to index. In *ACM Multimedia '98*.

Lemur Toolkit (2001). http://www-2.cs.cmu.edu/~lemur.

M Lesk and G Salton (1969). Relevance assessments and retrieval systems evaluation. *Information Storage and Retrieval 4*, 343–359.

D J Levitin (2001). Memory for musical attributes. In P Cook (Ed), *Music Cognition and Computerised Sound: An Introduction to Psychoacoustics*, pp 209–227. MIT Press.

C C Liu, J L Hsu and A L P Chen (1999a). An approximate string matching algorithm for content-based music retrieval. In *IEEE International Conference on Multimedia Computing and Systems*.

C C Liu, J L Hsu and A L P Chen (1999b). Efficient theme and non-trivial repeating pattern discovering in music databases. In *IEEE Intl. Conf. on Data Engineering*, pp 14–21.

G Lu (1999). *Multimedia Database Management Systems*. Artech House.

K MacMillan, M Droettboom and I Fujinaga (2001). Gamera: A structured document recognition application development environment. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 15–16.

K D Martin (1996). A blackboard system for automatic transcription of polyphonic music. Technical Report No. 385, MIT Media Laboratory.

R McNab, L A Smith, D Bainbridge and I H Witten (1997, May). The New Zealand digital library MELody inDEX. *D-Lib Magazine*.

R J McNab, L A Smith, I H Witten, C L Henderson and S J Cunningham (1996). Towards the digital music library: Tune retrieval from acoustic input. In *First ACM International Conference on Digital Libraries, DL '96*, pp 11–18.

C T Meadow, B B Boyce and D H Kraft (2000). *Text Information Retrieval* (2 ed). Academic Press.

C Meek and W P Birmingham (2001). Thematic extractor. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 119–128.

D Meredith, K Lemström and G Wiggins (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research 31*(4).

M Mongeau and D Sankoff (1990). Comparison of musical sequences. *Computers and the Humanities 24*, 161–175.

B Mont-Reynaud and M Goldstein (1985). On finding rhythmic patterns in musical lines. In *ICMC '85*, pp 391–387.

G Myers (1998). A fast bit-vector algorithm for approximate string matching based on dynamic programming. In *Combinatorial Pattern Matching, CPM 98*.

G Nagler (1998). GN MIDI solutions. http://www2.iicm.edu/Cpub.

P Ogilvie and J Callan (2001, November). Experiments using the Lemur Toolkit. In *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*, Gaithersburg,Maryland, pp 103–108.

T A Olson and S J Downie (2003). Chopin early editions: Construction and usage of online digital scores. In *Fourth International Conference on Music Information Retrieval,*

ISMIR 2003, pp 247–248.

D Parsons (1975). *The Directory of Tunes and Musical Themes*. Cambridge: Spencer Brown.

J Pickens (2000). A comparison of language modeling and probabilistic text information retrieval. In *1st Annual International Symposium on Music Information Retrieval, ISMIR2000*.

J Pickens, J P Bello, G Monti, T Crawford, M Dovey, M Sandler and D Byrd (2002). Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling approach. In *3rd International Conference on Music Information Retrieval, ISMIR 2002*, pp 140–149.

J Pickens and T Crawford (2002). Harmonic models for polyphonic music retrieval. In *Conference on Information and Knowledge Management, CIKM '02*, pp 438–445.

M D Plumbley, S Abdallah, J P Bello, M E Davies, G Monti and M B Sandler (2002). Automatic music transcription and audio source separation. *Cybernetics and Systems 33*(6), 603–627.

J M Ponte and W B Croft (1998). A language modeling approach to information retrieval. In *21st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp 275–281.

L Prechelt and R Typke (2001). An interface for melody input. *ACM Transactions on Computer-Human Interaction 8*(2), 133–149.

D Pye (2000). Content-based methods for the management of digital music. In *International Conference on Audio, Speech and Signal Processing, ICASSP 2000*.

C Raphael (2001). Automated rhythm transcription. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*.

J Reiss, J Aucouturier and M Sandler (2001). Efficient multidimensional searching routines for music information retrieval. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 163–171.

C J van Rijsbergen (1979). Information retrieval. http://www.dcs.gla.ac.uk/Keith/Preface.html. Online book.

S Robertson (2000). *Lectures on Information Retrieval*, Chapter Evaluation in Information Retrieval, pp 81–92. Springer-Verlag.

S E Robertson, S Walker, S Jones, M M Hancock-Beaulieu and M Gatford (1994). Okapi at TREC-3. In *NIST Special Publications 500-225: Overview of the Third Text Retrieval Conference (TREC-3)*.

G Salton (1968). *Automatic Information Organisation and Retrieval*. McGraw-Hill.

G Salton (1989). *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley.

T von Schroeter (2000). Auto-regressive spectral line analysis of piano tones. Technical report, Department of Computing, Imperial College London.

E Selfridge-Field (Ed) (1997). *Beyond MIDI: The Handbook of Musical Codes*. MIT Press.

E Selfridge-Field (1998). Conceptual and representational issues in melodic comparison. *Computing in Musicology 11*, 1–64.

C E Shannon (1948, July, Oct). Mathematical theory of communication. *Bell Systems Technical Journal 27*, 379–423, 623–656. Part 1 and 2.

C E Shannon (1951). Prediction and entropy of printed English. *Bell Systems Technical Journal 30*, 50–64.

I Shmulevich, O Yli-Harja, E Coyle, D-J Povel and K Lemström (1999). Perceptual issues in music pattern recognition - complexity of rhythm and key finding. In *Proceedings of the AISB '99 Symposium on Musical Creativity*, pp 64–69.

L Smith (1997). SCORE. In E Selfridge-Field (Ed), *Beyond MIDI: The Handbook of Musical Codes*, pp 252–280. The MIT Press. SCORE is a registered trademark of San Andreas Press.

L Smith and R Medina (2001). Discovering themes by exact pattern matching. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 31–32.

T Sødring and A F Smeaton (2002). Evaluating a music information retrieval system - TREC style. In *Panel Discussion, 3rd International Conference on Music Information Retrieval, ISMIR2002*.

T Sonoda and Y Muraoka (2000). A WWW-based melody-retrieval system - an indexing method for a large melody database. In *ICMC 2000, Berlin*, pp 170–173.

K Spärck Jones (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*(1), 11–20.

K Spärck Jones and P Willett (1997). *Readings in Information Retrieval*, Chapter Overall Introduction, pp 1–7. Morgan Kaufmann.

Amanda Spink and Howard Greisdorf (2001). Regions and levels: Measuring and mapping users' relevance judgments. *Journal of the American Society for Information Science and Technology 52*(2), 161–173.

David A Stech (1981). A computer-assisted approach to micro analysis of melodic lines. *Computers and the Humanities 15*, 211–221.

D M Sunday (1990, August). A very fast substring search algorithm. *Communications of the ACM 33*(8), 132–142.

J Tague-Sutcliffe (1992). An introduction to informetrics. *Information Processing and Management 28*(1), 1–3.

J Tague-Sutcliffe (1997). The pragmatics of information retrieval experimentation, revisited. In K Spärck Jones and P Willett (Eds), *Readings in Information Retrieval*, pp 205–216. Morgan Kaufmann Publishers, Inc.

A Tarter (2003). Query by humming. Technical report, Department of Computing, Imperial College London.

Y Tseng (1999). Content-based retrieval for music collections. In *SIGIR '99*, pp 176–182.

G Tzanetakis (2003). MARSYAS: Manipulation, analysis, and retrieval systems for audio and signals. http://sourceforge.net/projects/marsyas.

G Tzanetakis, G Essl and P Cook (2001). Automatic musical genre classification of audio signals. In *2nd International Symposium on Music Information Retrieval, ISMIR 2001*, pp 205–210.

A Uitdenbogerd (2002). *Music Information Retrieval Technology*. PhD thesis, Royal Melbourne Institute of Technology.

A Uitdenbogerd and J Zobel (1999). Melodic matching techniques for large databases. In *ACM Multimedia '99*, pp 57–66.

E Ukkonen (1992). Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 191–211.

B Vickery (1994). *Fifty years of information progress*, Chapter Introduction, pp 1–14. A Journal of Documentation Review. Aslib.

E M Voorhees (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management 36*, 697–716.

E M Voorhees (2001, September). Evaluation by highly relevant documents. In *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001*, pp 74–81.

E M Voorhees (2002). Wither Music IR evaluation infrastructure: Lessons to be learned from TREC. In *Workshop on MIR Evaluation, JCDL 2002*.

E M Voorhees and C Buckley (2002, August). The effect of topic set size on retrieval experiment error. In *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, pp 316–323.

E M Voorhees and D K Harman (1999). Overview of the eighth text retrieval conference (TREC-8). In *NIST Special Publication 500-240, The Eighth Text REtrieval Conference TREC-8*, pp 1–14.

K Walters (2001). Music retrieval. Technical report, Department of Computing, Imperial College London.

A Wieczorkowska (2000). Towards musical data classification via wavelet analysis. In *12th International Symposium on Methodologies for Intelligent Systems, ISMIS 2000*.

I H Witten, Alistair Moffat and Timothy C Bell (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd Edition*. Morgan Kaufmann.

E Wold, T Blum, D Keislar and J Wheaton (1996). Content-based classification, search and retrieval of audio. *IEEE Multimedia 3*(3), 27–36.

D Wolfram (1992). Applying informetric characteristics of databases to IR system file design. *Information Processing and Management 38*(1), 121–133.

S Wu and U Manber (1992, October). Text searching allowing errors. *Communications of the ACM 35*(10), 83–91.

C Zhai (2001). Notes on the Lemur TFIDF model. http://www-2.cs.cmu.edu/ lemur/1.9/tfidf.ps. Unpublished Report.